UNIVERSITY OF DELHI

CNC-II/093/1/EC-1275/25/3
Dated: 31.07.2025

NOTIFICATION

Sub: Amendment to Ordinance V

(ECR 24-13/ dated 12.07.2025)

Following addition be made to Annexure-II-A to the Ordinance V (2-A) of the Ordinances of the University;

Add the following:

The syllabi of following Departments under Faculty of Mathematical Science based on Postgraduate Curriculum Framework 2024 are notified herewith for the information of all concerned :

Department	Syllabi	Annexure
Computer Science	1. M.Sc. Computer Science	1
	2. Master of Computer Applications (MCA)	2
Operational Research	Master of Operational Research (MOR)	3
Statistics	M.Sc. Statistics	4
Mathematics	M.Sc. Mathematics	5

REGISTRAR

Annexure-1

MASTER OF COMPUTER SCIENCE
2-YEAR FULL-TIME PROGRAMME

Post Graduate Curriculum Framework

under

NEP 2020

(w.e.f 2025)

Department of Computer Science
Faculty of Mathematical Sciences
University of Delhi
Delhi-110007

MSc Computer Science Programme Details:

- **Affiliation**

  The proposed programme shall be governed by the Department of Computer Science, Faculty of Mathematical Sciences, University of Delhi, Delhi-110007.

- **Programme Structure and Objectives**

  The M.Sc. Computer Science programme is a four-semester program spanning two years. It has three structures - **MSc Computer Science with Coursework, MSc Computer Science with Coursework and Research, and MSc Computer Science with Research.** The first year (Semesters I and II) are common to all three structures. In the second year, the student can choose the structure he/she wish to follow.

  The Programme objectives of the M. Sc Computer Science with coursework are to

  - Equip the students with comprehensive knowledge of the current trends in computer science.
  - Enable the students to follow the career path of their choice by choosing courses from a wide list of specialised courses with progression.
  - Prepare the students to take up a career in the highly competitive **IT industry with development skills.**

  The Programme objectives of M. Sc Computer Science with Coursework and Research are to

  - Equip the students with comprehensive knowledge of the current trends in computer science and **introduce them to computer science research.**
  - Enable the students to follow the career path of their choice by choosing courses from a wide list of specialised courses with progression.
  - Prepare the students to take up a career
    - in the highly competitive IT industry **with research/ development skills**.
    - computer science research.

  The Programme objectives of M. Sc Computer Science with Research are to

  - Equip the students with comprehensive knowledge of the current trends in **computer science research**.
  - Enable the students to follow the research path of their choice by choosing courses from a wide list of specialised courses with progression.
  - Prepare the students to take up a career in
    - the highly competitive IT industry **with research**
    - computer science research.

Semester I

Semester I					
	Number of core courses				3
Course Code	**Course Title**	**Credits in each core course**			
		Theory	Tutorial	Practical	**Total**
DSC101	Beyond Polynomial-Time-Algorithms	3	0	1	**4**
DSC102	Machine Learning	3	0	1	**4**
DSC103	Mathematical Foundations of Computer Science	3	0	1	**4**
SBC101	Scientific Writing and Computational Analysis Tools	0	0	2	**2**
	Total credits in core course				14
	Number of DSE/GE courses				2*
	DSE1	3	0	1	**4**
	DSE2/ GE1	3	0	1	**4**
	Total credits in DSE/GE				8
	Total credits in Semester I				**22**

***Select two DSEs or one DSE and one GE**

List of DSEs for Semester I		
Course Code	**Course Title**	**L-T-P**
DSE101	Network Science	3-0-1
DSE102	Distributed and Parallel Computing	3-0-1
DSE103	Internetworking with TCP/IP	3-0-1
DSE104	Internet of Things	3-0-1
DSE105	Graph Theory	3-0-1
DSE106	Soft Computing	3-0-1

List of GEs for Semester I		
Course Code	**Course Title**	**L-T-P**
GE101	Data Analysis and Visualization	3-0-1
GE102	Programming with Python	3-0-1

Semester II					
	Number of core courses				3
Course Code	**Course Title**	**Credits in each core course**			
		Theory	Tutorial	Practical	**Total**
DSC201	Information Security	3	0	1	**4**
DSC202	Deep Learning	3	0	1	**4**
DSC203	Wireless and Mobile Communications	3	0	1	**4**
SBC201	Ethics for Responsible AI	1	0	1	**2**
	Total credits in core course				14
	Number of DSE/GE courses				2*
	DSE3	3	0	1	**4**
	DSE4/GE2	3	0	1	**4**
	Total credits in DSE/GE				8
	Total credits in Semester II				**22**

***Select two DSEs or one DSE and one GE**

List of DSEs for Semester II		
Course Code	**Course Title**	**L-T-P**
DSE201	Social Networks Analysis	3-0-1
DSE202	Combinatorial Optimization	3-0-1
DSE203	Cyber Security	3-0-1
DSE204	Information Retrieval	3-0-1
DSE205	Digital Image Processing	3-0-1
DSE206	Advanced Classification Methods	3-0-1
DSE207 (Only for structure 3)	Computer Vision	3-0-1

List of GEs for Semester II		
Course Code	**Course Title**	**L-T-P**
GE201	Data Mining	3-0-1
GE202	Data Science using Python	3-0-1

Semester III

Structure 1 Coursework

Semester III (Structure 1 Coursework)					
	Number of core courses				2
Course Code	**Course Title**	**Credits in each core course**			
		Theory	Tutorial	Practical	**Total**
DSC301	Optimization Methods	3	0	1	**4**
DSC302	Natural Language Processing	3	0	1	**4**
SBC301	Skill	0	0	2	**2**
	Total credits in core course				10
	Number of DSE/GE courses				3**
	DSE5	3	0	1	**4**
	DSE6	3	0	1	**4**
	DSE7/GE3	3	0	1	**4**
	Total credits in DSE/GE				12
	Total credits in Semester III				**22**

**** Select three DSEs or two DSEs and one GE**

List of DSE for Semester III (Structure 1 Coursework)		
Course Code	**Course Title**	**L-T-P**
DSE301	Special Topics in Theoretical Computer Science	3-0-1
DSE302	Link Prediction	3-0-1
DSE303	Recommender System	3-0-1
DSE304	Time Series Data Analysis	3-0-1
DSE305	Quantum Computing and Applications	3-0-1
DSE306	Digital Forensic and Incident Response	3-0-1
DSE307	Human Computer Interaction	3-0-1
DSE308	Influence Maximization	3-0-1

Semester IV (Structure 1 Coursework)					
	Number of core courses				2
Course Code	**Course Title**	**Credits in each core course**			
		Theory	Tutorial	Practical	**Total**
DSC401	Reinforcement learning	3	0	1	**4**

DSC402	Computer Vision	3	0	1	**4**
SBC401	Skill	0	0	2	**2**
	Total credits in core course				10
	Number of DSE/GE courses				3**
	DSE8	3	0	1	**4**
	DSE9	3	0	1	**4**
	DSE10/GE4	3	0	1	**4**
	Total credits in DSE/GE				12
	Total credits in Semester IV				**22**

**** Select three DSEs or two DSEs and one GE**

Structure 2 Course Work and Research

Semester III (Structure 2 Coursework and Research)					
	Number of core courses				2
Course Code	**Course Title**	**Credits in each core course**			
		Theory	Tutorial	Practical	**Total**
DSC301	Optimization Methods	3	0	1	**4**
DSC302	Natural Language Processing	3	0	1	**4**
	Research-1	0	0	6	**6**
	Total credits in core course				14
	Number of DSE/GE courses				2*
	DSE5	3	0	1	**4**
	DSE6/GE3	3	0	1	**4**
	Total credits in DSE/GE				8
	Total credits in Semester III				**22**

*** Select two DSEs or one DSE and one GE**

List of DSE for Semester III (Structure 2 Coursework and Research)		
Course Code	**Course Title**	**L-T-P**
DSE301	Special Topics in Theoretical Computer Science	3-0-1
DSE302	Link Prediction	3-0-1
DSE303	Recommender System	3-0-1
DSE304	Time Series Data Analysis	3-0-1
DSE305	Quantum Computing and Applications	3-0-1
DSE306	Digital Forensic and Incident Response	3-0-1
DSE307	Human Computer Interaction	3-0-1
DSE308	Influence Maximization	3-0-1

*** Select two DSEs or one DSE and one GE**

Semester III (Structure 3 Research)					
	Number of core courses	1			
Course Code	**Course Title**	**Credits in each core course**			
		Theory	Tutorial	Practical	**Total**
DSC301	Optimization Methods	3	0	1	**4**
	Advanced Research Methodology	1	0	1	**2**
	Tools for Research	0	0	2	**2**
	Research-1	0	0	10	**10**
	Total credits in core course	18			
	Number of DSE courses	1			
	DSE5	3	0	1	**4**
	Total credits in DSE	4			
	Total credits in Semester III	**22**			

Semester IV (Structure 2 Coursework and Research)					
	Number of core courses	2			
Course Code	**Course Title**	**Credits in each core course**			
		Theory	Tutorial	Practical	**Total**
DSC401	Reinforcement learning	3	0	1	**4**
DSC402	Computer Vision	3	0	1	**4**
	Research-2	0	0	6	**6**
	Total credits in core course	14			
	Number of DSE/GE courses	2*			
	DSE7	3	0	1	**4**
	DSE8/GE4	3	0	1	**4**
	Total credits in DSE/GE	8			
	Total credits in Semester IV	**22**			

Structure 3 Research

List of DSE for Semester III (Structure 3 Research)		
Course Code	**Course Title**	**L-T-P**
DSE301	Special Topics in Theoretical Computer Science	3-0-1
DSE302	Link Prediction	3-0-1
DSE303	Recommender System	3-0-1
DSE304	Time Series Data Analysis	3-0-1
DSE305	Quantum Computing and Applications	3-0-1
DSE306	Digital Forensic and Incident Response	3-0-1
DSE307	Human Computer Interaction	3-0-1
DSE308	Influence Maximization	3-0-1
DSE309	Natural Language Processing	3-0-1

Semester IV (Structure 3 Research)					
	Number of core courses				0
Course Code	**Course Title**	**Credits in each core course**			
		Theory	Tutorial	Practical	**Total**
	Techniques for Research Writing	1	0	1	**2**
	Research-2	0	0	16	**16**
	Total credits in core course				18
	Number of DSE courses				1
		Theory	Tutorial	Practical	**Total**
	DSE6	3	0	1	**4**
	Total credits in DSE				4
	Total credits in Semester IV				**22**

List of DSE for Semester IV (To be decided)		
Course Code	**Course Title**	**L-T-P**

SEMESTER - I

DSC101: Beyond Polynomial-Time-Algorithms [3-0-1]

Course Code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/Practice	
DSC101	**4**	**3**	**0**	**1**	**A UG level course in Design and Analysis of Algorithms**

Course Objectives

This course is designed to introduce some problems that are faced in real life but are too hard to admit fast solutions. Reduction technique to prove the hardness of a problem is discussed. Some techniques that give fast approximate solutions to these problems are also explored. The solutions provide provable guarantees on the cost relative to the optimum.

Course Learning Outcomes:

Upon successful completion of this course, students will

1. be able to appreciate that certain problems are too hard to admit fast solutions and be able to prove their hardness.
2. be able to explain what an approximation algorithm is, and the advantage of using approximation algorithms.
3. be familiar with some techniques to design approximation algorithms.
4. be familiar with some techniques to analyse the approximation factor of an algorithm.

Syllabus

Unit I (11 hours)

Introduction to Classes P, NP, NP-Hard, NP-Complete: Verifiability and Reduction. Constraint Satisfaction: Satisfiability Problem (SAT), 3SAT.

Unit II (9 hours)

Graph and Set Problems: Clique, Vertex Cover, Independent Set, Hamiltonian Cycle Problem, Travelling Salesman Problem. Sets and Partitions: Set partition, Set Cover, Subset Sum and Knapsack Problem.

Unit III (10 hours)

NP Hardness of Clustering Problems: k-means clustering, k-centre clustering and k-median clustering. Machine Learning Algorithms: Introduction to Clustering Algorithms, learning the predictors and fitting a model to data using gradient descent.

Unit IV **(15 hours)**

Techniques to design approximation algorithms: Introduction to Linear Programming and Integer Programming. Introduction to LP-rounding, Primal-Dual, Dual Fitting, Greedy and Local Search techniques to design approximation algorithms.

Readings:

1. Kleinberg, J. and Tardos,E. *Algorithm Design*. 1st Edition, Pearson Education India, 2013.
2. Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, *Introduction to Algorithms*, 4th Edition, The MIT Press, 2022.
3. Vazirani, Vijay V., *Approximation Algorithms*, Springer, 2013.
4. Williamson, David P., and David B. Shmoys. *The Design of Approximation Algorithms,* Cambridge University Press, 2011.

Practicals

General Instructions:

1. All programs to be developed/uploaded immediately on Github.
2. Data sets (synthetic/real) created for Practical 1 are to be used for Practical 2 and 4 also. So save them.
3. Data sets (synthetic/real) created for Practical 3 are to be used for Practical 4, 5 also. So save them.
4. Results obtained in a practical will be required in subsequent practicals. So save them.
5. A program must be completely automated with no manual intervention - from taking the input values like n, m to generating the records of the results. Generating data sets can be internal to the program. It will be better to save it in a file also, in case it is required to be re-used.
6. Time should not include the time for reading the input and writing the output.
7. If GenAI tools are used to create a part of the code, prior permission must be sought in writing and it must be recorded as documentation at the top of the program.
8. Libraries for Sorting, Searching, basic data structures like Stacks, Queues, Binary Search trees, Hashmaps, basic graph algorithms like DFS, BFS, MST (Prim's Kruskal), Djikstra's algorithm can be used. For any other specialised part, prior permission must be sought.
9. Standard LP solvers are to be used to solve Linear Programs.
10. For synthetic data sets:
    a. For graph problems:
        i. Create at least 10 data sets randomly for each pair of (n, m) where n is the number of vertices and m is the number of edges..

n	m	
10	10, 20, …100	
20	20, 40 …..100, 200, 300, 400	

List of Practicals

1. Implement/Use the brute force algorithm for the Vertex Cover problem for n = 10, m = 10, 20, …100. Record the size of the computed vertex cover and the running time for each (n, m) pair. **(10 hours)**

2. Implement/Use the greedy 2-factor approximation algorithm for the Vertex Cover problem for n = 10, m = 10, 20, …100 **(4 hours)**
   a. Record the size of the computed vertex cover and the running time for each (n, m) pair.
   b. Compute the Approximation factor for each (n, m) pair using the results obtained in Practical 1 above.

3. For n = 20 for each data set: **(2 hours)**
   a. Run the brute force algorithm for the Vertex Cover problem. Record the size of the computed vertex cover and the running time.
   b. Run the greedy approximation algorithm for the Vertex Cover problem. Record the size of the computed vertex cover and the running time
   c. Compute the Approximation factor of greedy by comparing the results with those of brute force.

4. Write a program to approximate Vertex Cover by 2 factor using LP- rounding and run it for n = 10, 20. **(4 hours)**
   For each (n, m) pair, compute the Approximation factor of greedy algorithm (obtained in Practical 2 and 3(b)) by comparing the results with LP optimal.

5. Take 5 real data sets. For each data set: **(5 hours)**
   a. Use a library and run a (heuristic)algorithm for the k-centre problem. Record the cost and the time. Run on toy data first.
   b. Implement/Use and run a greedy 2-approximation algorithm for the k-centre algorithm. Record the cost and the time, and compare with the observations in part (a). Run on toy data first.

6. As per the recent development on approximation algorithms for clustering problems. For instance, a recent approximation algorithm for k-centre with outliers. **(5 hours)**

DSC102: Machine Learning [3-0-1]

Course Code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/Practice	
DSC 102	**4**	**3**	**0**	**1**	**Graduation**

Course Objectives:

This course provides a foundational understanding of machine learning (ML) concepts, covering both supervised and unsupervised learning techniques. It aims to equip students with the knowledge required to develop, analyze, and evaluate ML models. Emphasis is placed on practical applications and the theoretical underpinnings of ML algorithms.

Course Learning Outcomes

Upon successful completion of this course, students will be able to

1. Understand fundamental concepts of machine learning, including hypothesis space, bias-variance trade-off, dimensionality reduction, and model evaluation metrics.
2. Design and implement regression and classification models such as linear regression, logistic regression, decision trees, SVMs, and neural networks using real-world datasets.
3. Apply clustering techniques such as k-means, hierarchical, and density-based clustering, and evaluate the quality of clustering results.
4. Perform model tuning and selection using appropriate validation techniques and performance metrics.

Syllabus

Unit-I **(10 hours)**

Introduction to Machine Learning: Hypothesis and target class, bias-variance tradeoff, Occam's razor, approximation and estimation errors, curse of dimensionality, dimensionality reduction, feature scaling, feature selection methods.

Unit-II **(10 hours)**

Regression: Linear regression with one variable, linear regression with multiple variables, gradient descent, logistic regression, polynomial regression, over-fitting, regularization. performance evaluation metrics, validation methods.

Unit-III **(12 hours)**

Classification: Decision trees, Naive Bayes classifier, support vector machine, kernel functions, perceptron, multilayer perceptron, neural network, back-propagation algorithm, validation methods.

Unit-IV **(13 hours)**

Clustering: Introduction, clustering paradigms, similarity and distance, density, characteristics of clustering algorithms, centre-based clustering techniques, hierarchical clustering, density-based clustering, other clustering techniques, scalable clustering algorithms, methods for clustering validation.

Readings

1. Alpaydin, Ethem. *Introduction to machine learning*. MIT press, 2020.
2. Mitchell, Tom M., and Tom M. Mitchell. *Machine learning*. Vol. 1, no. 9. New York: McGraw-hill, 1997.
3. Bishop, Christopher M., and Nasser M. Nasrabadi. *Pattern recognition and machine learning*. Vol. 4, no. 4. New York: springer, 2006.

4. Shalev-Shwartz, Shai, and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.
5. Michalski, Ryszard S. *Machine learning: An artificial intelligence approach*. Springer Science & Business Media, 2013.

List of Practicals :

1. Perform feature engineering and selection on a dataset by applying feature scaling techniques such as Min-Max scaling and standardisation, using PCA for dimensionality reduction, and implementing at least two feature selection methods like correlation-based filtering and chi-square test. **(4 hours)**
2. Simulate the bias-variance trade-off using polynomial regression on synthetic data and explore the impact of high-dimensional feature spaces on model performance using PCA. **(4 hours)**
3. Implement linear regression from scratch using one and multiple variables, and visualise and interpret the resulting models. **(2 hours)**
4. Implement logistic regression to classify binary data, demonstrate the effect of overfitting, and apply L1 and L2 regularisation techniques. **(2 hours)**
5. Evaluate regression models using performance metrics such as MSE, RMSE, and R^2, and apply k-fold cross-validation to assess the generalisation ability of the models. **(2 hours)**
6. Implement decision trees and Naive Bayes classifiers on a real dataset, and visualise the resulting decision boundaries and tree structures. **(2 hours)**
7. Train support vector machine (SVM) models using both linear and non-linear (e.g., RBF) kernel functions, and visualise the resulting classification boundaries. **(4 hours)**
8. Build a neural network model using a small dataset using existing libraries, and explain and visualise the learning process, including weights and activations. **(4 hours)**
9. Apply k-means and hierarchical clustering on a dataset such as customer segmentation, visualise the resulting clusters, and validate the results using silhouette scores. **(4 hours)**
10. Apply the DBSCAN clustering algorithm to a dataset with non-globular clusters, compare its performance with k-means, and visualise the differences in clustering behaviour. **(2 hours)**

DSC103: Mathematical Foundations of Computer Science [3-0-1]

Course Code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/Practice	
DSC103	**4**	**3**	**0**	**1**	**Graduation**

Course Objectives:

This course will discuss fundamental concepts and tools in discrete mathematics with emphasis on their applications to computer science. The objectives of this course comprise providing students with knowledge of logic and Boolean circuits, sets, functions, relations, deterministic and randomised algorithms. Furthermore, the students will learn analysis techniques based on counting methods, recurrence relations, trees and graphs.

Course Learning Outcomes:

At the end of the course, students will be able to

1. perform operations on vectors; represent vectors geometrically; apply vector algebra to solve problems in sub-disciplines of computer science.
2. perform operations on matrices and sparse matrices; compute the determinant, rank and eigenvalues of a matrix; apply matrix algebra to solve problems in sub-disciplines of computer science.
3. perform data analysis in a probabilistic framework.
4. visualise and model the given problem using mathematical concepts covered in the course.

Syllabus:

Unit-I **(8 Hours)**

Vectors: Definition of Vectors, Vector Addition, Dot and Cross Products, Span, Norm of vectors, Orthogonality, geometry of vectors, Application of vectors in document analysis.

Unit-II **(12 Hours)**

Matrix Algebra: Matrices as vectors; Matrix-vector, vector-matrix and matrix-matrix multiplications; Inner and outer products, triangular matrix, diagonal matrix, systems of linear equations, linear independence, determinant, rank of matrix, Eigen values and Eigen vectors, matrix transformations, geometry of transformations, Applications of matrix algebra in image representation and transformations.

Unit-III **(11 Hours)**

Probability Theory and Basic Statistics: Sample Space and Events, Probability axioms, Conditional Probability, Bayes' law, Introduction to Descriptive and Inferential Statistics, Describing Data Sets as Frequency tables, Relative frequency tables and graphs, Scatter diagram, Grouped data, Histograms, Ogives; Percentiles, Box Plot, Coefficient of variation, Skewness, Kurtosis.

Unit-IV **(14 Hours)**

Distributions: Continuous and Discrete random variables, probability density function, probability mass function, distribution function and their properties, mathematical expectation, conditional expectation, Uniform (continuous and discrete), Binomial, Poisson, Exponential, Normal, $\chi 2$ distributions, weak Law of Large Numbers, Central Limit Theorem, Chebyshev's inequality. Stochastic Processes Introduction to stochastic process, Markov Chain, Transition probabilities, Birth-Death process

Readings:

1. Trivedi, Kishor S, Probability & statistics with reliability, queuing and computer science applications. John Wiley & Sons, 2008.
2. Ross Sheldon M, Probability models for computer science. Academic Press, 2001.
3. Davis, Ernest. Linear algebra and probability for computer science applications. CRC Press, 2012.

4. Norm Matloff, From Algorithms to Z-Scores: Probabilistic and Statistical Modeling in Computer Science, University of California, Davis (Creative CommonLicence)https://www.cs.ucdavis.edu/~matloff/matloff/public_html/ProbStatBook.pdf

List of Practicals :

1. Write a modular program to create a term-frequency matrix using the extracted text. Each row of the matrix corresponds to a unique term (word) and each column represents one of the selected paragraphs. The value in each cell of the matrix should indicate how frequently the term appears in the respective paragraph. With this matrix, compute two separate paragraph similarity matrices of size n × n: one using cosine similarity and the other using dot product. These matrices will capture the pairwise similarity between all paragraphs based on their word usage patterns**. (4 hours)**

   a) Once the two similarity matrices are generated, analyze the results to determine which two paragraphs are the most similar and which two are the most dissimilar under each method. Compare the findings from the cosine similarity matrix and the dot product matrix to see if they highlight the same paragraph pairs as being similar or different.
   b) After completing this analysis, prepare a brief report (no more than three pages) summarizing your observations. The report should clearly state the most and least similar paragraph pairs according to both methods, and whether the results align across similarity measures. Additionally, identify the five terms that contribute most significantly to the similarity and dissimilarity between the selected paragraphs. Conclude the report with insights into why these terms might be influential and any notable patterns observed during the comparison.

2. Take a picture with your phone camera, transfer it to your computer without applying any filters or edits. Read the image file and convert it to a grayscale format. Resize the image to a manageable dimension to suit the specifications of your machine (maintain the aspect ratio). Save the grayscale image as a 2D matrix. Design a menu-driven application that allows the user to choose from the following geometric transformations: rotation, scaling, and translation. Prompt the user to enter the required parameters (angle for rotation, scaling factor, translation distances) for each transformation, and validate input appropriately. Display the original grayscale image alongside the transformed versions accompanied by a descriptive caption indicating the type of transformation and parameters used. Also, write technical specifications of the camera and compile the report. **(4 hours)**

3. Download a dataset consisting of N grayscale human face images. **(8 hours)**

I. Design a modular program to perform the following:

   a) Read each image, convert it to grayscale if needed, and resize it to 32 × 32 pixels for uniformity.
   b) Flatten each image into a 1D vector of size 1024 (i.e., 32 × 32)
   c) Stack all vectors to form a 2D array of shape N × R, where R = 1024. In this array, each row corresponds to a different face image in vector form.
   d) Apply Principal Component Analysis (PCA) to the matrix and retain only the top eigenvectors that explain at least 95% of the total variance.
   e) Reshape the top 5 eigenvectors into 32 × 32 images and plot them. (Note: These visualisations are called eigenfaces, and they represent the most significant facial features in the dataset.)

II. Next, take a selfie using your smartphone, ensuring that your face is well-lit, in focus, and centred. Transfer this selfie to your computer, convert it to grayscale, and resize it to 32 × 32 pixels. Display the image, then project it onto the eigenface space using the eigenfaces extracted in step (d). Reconstruct and render your face using a linear combination of the top eigenfaces, and display the result side-by-side with the original for comparison.

III. Start with N=50, and if your system resources allow, increase the dataset to 100 faces and, if possible, resize all images, including your selfie, to 64 × 64 pixels. Apply PCA again to this higher-resolution dataset and plot the top 5 eigenfaces. Resize and project your face similarly, then display the reconstructed result.

IV. Finally, compile your observations into a report. Place at most two pairs of images per page (e.g., original vs reconstructed), side by side, and add brief, insightful observations beneath each pair. These observations should compare the quality of reconstruction, differences between 32×32 and 64×64 results, and how well the eigenfaces capture facial features.

4. Solving systems of linear equations. Compute rank, determinant, eigenvalues and eigenvectors of a matrix. **(2 Hours)**

5. Central limit theorem demonstration: Simulate sums/averages of random samples and show convergence to normal distribution. **(2 Hours)**

6. Simulating 1D and 2D random walks for a stochastic process, plot multiple paths, compute average position and variance over time. **(2 Hours)**

7. Construct and validate the Markov Chain Transition probability matrix on real-world examples (e.g., weather: Sunny, Rainy). **(2 Hours)**

8. Compute n-step transition probabilities to find long-term transitions: Define transition matrix P. Compute P^2, P^3, ... P^n using matrix multiplication. Show the probability of moving from state i to j in n steps. **(2 Hours)**

9. Stationary Distribution of a Markov Chain: Identify steady-state behaviour. Solve: $\pi P = \pi$, subject to sum(π) = 1. Simulation for frequencies converges to π. Analyse the structure of the transition graph. Check if all states are reachable or not. Check periodicity and irreducibility. **(2 Hours)**

10. Simulate the birth-death process for a queue or population models M/M/1 or M/M/M//∞, like customer service (bank, call centre, etc.), network packet queues, and hospital patients (arrival/departure) systems. Calculate birth rate (λ), rate (μ), and other performance metrics. **(2 Hours)**

SBC101: SCIENTIFIC WRITING AND COMPUTATIONAL ANALYSIS TOOLS [0-0-2]

Course Code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice	
SBC101	2	0	0	2	Graduation

Course Objective:

The objective of this course is to develop proficiency in the use of software tools required for project development.

Course Learning Outcomes:

Upon successful completion of this course, students will be able to

1. create a basic LaTeX document, format text with different styles, and apply fundamental document structures such as sections, paragraphs, and formatting options.

2. create and manipulate matrices and arrays in MATLAB, understanding their importance in numerical computation and data analysis.

3. apply optimisation techniques and perform curve fitting tasks in MATLAB, including fitting data to a model and minimizing error.

Syllabus:

Unit-I

LaTex : Basic Document Setup, and Formatting Text, Lists and Tables, Multi-Column Layouts, Mathematical Equations, Figures and Graphics, Customizing Fonts and Colors, References, Citations, and Headers/Footers/Page Numbering, Creating a Resume or CV.

Unit-II

MATLAB: Basic MATLAB Operations, Matrices and Arrays, Plotting and Visualization, Loops and Conditional Statements, Functions and Scripts, File Handling, Signal Processing, Image Processing, Optimization and Curve Fitting, Simulink Basics

Readings:

1. https://guides.nyu.edu/LaTeX/sample-document
2. Brian R. Hunt, Ronald L. Lipsman, Jonathan M. Rosenberg. *A Guide to MATLAB: For Beginners and Experienced Users*, Cambridge University Press, 2017.

List of Practicals

1. Basic Document Setup and Formatting Text: Create a new document with a specific page size, orientation, and margins, Format a paragraph using different font styles (bold, italic, underline) and sizes, Apply alignment options (left, right, center, justified) to different text blocks, Use styles to format headings and body text consistently. **(4 hours)**

2. Lists and Tables, Multi-Column Layouts: Create a bulleted and numbered list for an agenda or shopping list, design a table to represent student grades or inventory details, merge and split cells in the table, and apply borders and shading. Create a newsletter-style document using two or three-column layouts.**(4 hours)**

3. Mathematical Equations: Insert and format basic math expressions, such as fractions, superscripts, and subscripts. Write complex equations, like the quadratic formula, matrices, and integrals, using equation tools or LaTeX syntax (if applicable), and align multiple equations properly using equation editors or tab alignment. **(4 hours)**

4. Figures and Graphics, Customising Fonts and Colours: Insert an image or figure into the document with a caption, Resize and position the image using wrap text and alignment options, Change font family and colour scheme for different document elements, Create a cover page with custom fonts, colours, and images. **(4 hours)**

5. References, Citations, and Headers/Footers/Page Numbering: Insert a bibliography and add citations using a reference manager or built-in citation tools, Apply consistent header and footer designs across the document, Insert and format page numbers (e.g., Roman numerals for intro, Arabic for content), Create a Table of Contents and update it automatically. **(4 hours)**

6. Creating a Resume or CV: Use a template or design a CV layout from scratch with appropriate sections (Education, Experience, Skills), Insert a profile photo, format contact details, and add social media links, Apply proper use of whitespace, bullets, and alignment to ensure clarity and readability, Export the CV as a PDF and check formatting consistency. **(4 hours)**

7. Introduction to MATLAB Interface and Basic Commands: Using command window, editor, and workspace, Arithmetic operations, using help, clc, clear, who, whos. Variable Assignment and Data Types: Creating scalars, vectors, complex numbers, Type conversion, and precision handling, Matrix Creation and Manipulation: Defining matrices, transposition, reshaping, Indexing, slicing, concatenation, Matrix multiplication, inversion, determinant, eigenvalues. **(4 hours)**

8. Plotting and Visualization 2D and 3D Plotting: *plot(), subplot(), title(), xlabel(), ylabel(), legend(), 3D plots: mesh(), surf(), contour()*. Data Visualisation with Customisation: Bar graphs, pie charts, histograms, Line styles, markers, colour changes. **(4 hours)**

9. Loops and Conditional Statements: Using for, while, and if-else, writing loops to sum a series, the Fibonacci series, and Conditionals to check prime numbers or grading systems, Functions and Scripts, Creating User-Defined Functions, writing a function to compute factorial, and standard deviation. **(4 hours)**

10. Using Scripts for Automation: Writing scripts to read input, process, and display output, File Handling: Reading and Writing Files, reading data from .txt, .csv using *fopen, fscanf, textscan, readmatrix()*. Writing output to files. **(4 hours)**

11. Signal Processing: Basic Signal Generation and Analysis, generating sine, square, and triangular waves, plotting signals, and performing FFT. Filtering Signals: Applying FIR and IIR filters, using filter (), butter (), and freqz(). **(4 hours)**

12. Image Processing: Image Reading and Display, Read and display an image using *imread(), imshow(), and rgb2gray()*. Image Enhancement and Filtering: Histogram equalization, edge detection, smoothing**. (8 hours)**

13. Optimisation and Curve Fitting: Curve Fitting using *polyfit()* and *fit()*. Fit a polynomial or custom function to data, Evaluate goodness of fit, optimisation using fminsearch, fmincon: Minimise a nonlinear function with constraints. **(8 hours)**

List of DSEs for Semester I

DSE101: NETWORK SCIENCE [3-0-1]

Course Code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice	

DSE101	**4**	**3**	**0**	**1**	**Basic knowledge of probability theory**

Course Objectives:

The course aims to acquaint the students with the graph theory concepts relevant for network science. The students learn the dynamics of and on networks in the context of applications from disciplines like biology, sociology, and economics

Course Learning Outcomes:

At the end of the course, students will be

1. able to appreciate ubiquity of graph data model.
2. able to understand the importance of graph theoretic concepts in social network analysis.
3. able to understand the structural features of a network.
4. familiar with the theoretical graph generation models.
5. identify community structures in networks.
6. able to write programs to solve complex network problems.

Syllabus:

Unit-I **(5 hours)**

Introduction: Introduction to complex systems and networks, modelling of complex systems, review of graph theory.

Unit II **(10 hours)**

Network properties: Local and global properties, clustering coefficient, All-pair-shortest path-based properties, centrality measures for directed and undirected networks, degree distribution, and clustering coefficient.

Unit III **(15 hours)**

Graph models: Random graph model, degree distribution, small world network model, power laws and scale-free networks, Barabasi-Albert (preferential attachment) network model, measuring preferential attachment.

Unit IV **(15 hours)**

Community structure in networks: Communities and community detection in networks, Hierarchical algorithms for community detection, Modularity-based community detection algorithms, label propagation algorithm, multi-level graph.

Readings:

1. Pósfai, Márton, and Albert-László Barabási, *Network Science*, Cambridge University Press, 2016.
2. Newman, MEJ. *Networks: An Introduction*, Oxford University Press, 2010.
3. Easley David, and Jon Kleinberg, *Networks, crowds, and markets: Reasoning about a highly connected world*, Cambridge university press, 2010.
4. Meira Jr, W. A. G. N. E. R., and M. J. Zaki, *Data mining and analysis, Fundamental Concepts and Algorithms*, 2014.

List of Practicals :

1. Create and Visualize a Simple Network and understand nodes, edges, and basic network structure. Use NetworkX (Python), Gephi **(4 hours)**
2. Visualize Networks Using User Attributes. Use node color, size, or layout to reflect attributes like age or interest. **(2 hours)**
3. Implement Local and Global Network Properties. Also, measure average path length, diameter, and network density. **(2 hours)**
4. Calculate Node Centrality Measures and compare degree, closeness, betweenness, and eigenvector centrality. Use social graph, e.g., Zachary's Karate Club and dolphin datasets. **(8 hours)**
5. Plot and analyze degree distributions in real-world and synthetic networks. **(2 hours)**
6. Implement Erdős–Rényi random graphs and assess their structural properties. **(2 hours)**
7. Generate Small World Networks using Watts-Strogatz model and measure clustering and path length. **(2 hours)**
8. Implement Barabási–Albert Model and Generate preferential attachment networks and observe scale-free properties. **(2 hours)**
9. Implement and apply Hierarchical Community Detection algorithms to discover community structures. **(4 hours)**
10. Use the Louvain algorithm to find communities and optimize communities using modularity. **(2 hours)**

DSE102: Distributed and Parallel Computing [3-0-1]

Course Code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/Practice	
DSE102	**4**	**3**	**0**	**1**	**Knowledge of Computer Architecture**

Course Objectives:

The course aims to provide comprehensive knowledge about Distributed and Parallel Computing. It includes understanding of Parallel Computing & Parallel Programming concepts, design and analysis of distributed & shared memory programming models, design and analysis of accelerated computing models, performance comparison of sequential and parallel programming models and mapping of applications to parallel computing systems.

Course Learning Outcomes:

On completing this course, students will be able to

1. understand the Parallel Computing and Parallel Programming concepts.
2. analyse the need for Parallel Computing.
3. design and analyse the shared memory programming models.
4. design and analyse the distributed memory programming models.
5. design and analyse the accelerated computing models.
6. demonstrate performance analysis of sequential and parallel programming models.
7. evaluate the performance of modern parallel computing systems.

Syllabus

Unit-I **(8 hours)**

Introduction: Introduction to Parallel Programming concepts, Parallel Architectures, Pipelining, Introduction to high-performance computing (HPC) and scientific computing. The Need for HPC. Processor performances, Memory hierarchy, multi-core processing and Vector computing. Models (SIMD, MIMD, SIMT, SPMD, Dataflow Models, Demand-driven Computation).

Unit-II **(7 hours)**

Performance Measures- speedup, execution time, efficiency, cost, scalability, effect of granularity on performance, scalability of parallel systems, Amdahl's Law, Gustafson's Law. Performance bottlenecks. Introduction to Linux for parallel programming, bash scripting. HPC cluster access and environment setup, applications of High-Performance Computing.

Unit-III **(15 hours)**

Shared and Distributed Memory Programming Models: Concept of Decomposition, Tasks, Dependency Graphs, Granularity, Decomposition Techniques, Characteristics of Tasks and Interactions, Mapping Techniques for Load Balancing, Parallel Algorithm Models. OpenMP and thread programming, Message Passing Interface (MPI), Introduction to GPGPU/Vector programming. Effective use of debuggers and parallel programming.

Unit-IV **(15 hours)**

Accelerated Computing models: GPU (Graphics Processing Unit), Introduction to GPU Architectures, Clock speeds, CPU / GPU comparisons, CUDA (Compute Unified Device Architecture), CUDA Programming, CUDA Terminology - Kernels, Threads, Blocks, Memory Management, Built-in Variables and Functions, Thread Scheduling, CUDA Memory Model, Thread Synchronization. GPU – OpenACC. Analyzing and Parallelising with OpenACC,

OpenACC Optimizations. Performance analysis of parallel programming. Use of toolkits such as BLAS, LAPACKz, PETSC. Advanced scientific visualisation.

<u>Readings:</u>

1. Pacheco, Peter. *An introduction to parallel programming*. Elsevier, 2011.
2. Kumar, Vipin, Ananth Grama, Anshul Gupta, and George Karypis. *Introduction to parallel computing*. Vol. 110. Redwood City, CA: Benjamin/Cummings, 1994.
3. Quin, M. *Parallel programming in C with MPI and OpenMP*. Tata McGraw Hills edition (2000).
4. Cook, Shane. *CUDA programming: a developer's guide to parallel computing with GPUs*. Newness, 2012.
5. Sanders, Jason. *CUDA by Example: An Introduction to General-Purpose GPU Programming*. Addison-Wesley Professional, 2010.

<u>References</u>

1. Hager, Georg, and Gerhard Wellein. *Introduction to high-performance computing for scientists and engineers*. CRC Press, 2010.
2. HPC Tutorials (https://hpc-tutorials.llnl.gov/).

List of Practicals:

1. Write Python Programs for Computing speedup using multiprocessors. **(2 hours)**
2. Installation of OpenMP environment setup **(4 hours)**
3. Using OpenMP, write a multithreaded program where each thread prints “hello world”. **(2 hours)**
4. Write a Parallel program that should print the series of 2 and 4. Make sure both should be executed by different threads. **(2 hours)**
5. Consider a scenario where you have to parallelise a program that performs matrix multiplication using OpenMP. Your task is to implement parallelization using both static and dynamic scheduling, and compare the execution time of each approach. **(4 hours)**
6. Consider a scenario where you have a shared variable total_sum that needs to be updated concurrently by multiple threads in a parallel program. However, concurrent updates to this variable can result in data races and incorrect results. Your task is to modify the program to ensure correct synchronization using OpenMP's critical and atomic constructs. **(4 hours)**
7. Write an MPI program to send and receive Hello World to a Root process and print the received messages. **(2 hours)**
8. Write an MPI program to send two numbers (array elements) per process to a Root process and print the received messages. **(2 hours)**
9. Write an MPI program to find the sum of ranks of all the processes. Implement using point-to-point communication as well as Collective communication routines. **(4 hours)**
10. Write an MPI program to parallelize serial code on Pi calculation. **(4 hours)**

DSE103: Internetworking with TCP/IP [3-0-1]

Course Code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice	
DSE103	**4**	**3**	**0**	**1**	**Knowledge of Computer Networks**

Course Objectives:

This course provides an in-depth understanding of the architecture, design and behaviour of the Internet through the TCP/IP suite of protocols. This course will enable students to test and troubleshoot IP-based communications systems. Furthermore, this course will explore various flow control and congestion control mechanisms of TCP, along with Quality of Service (QoS) techniques that enhance network performance.

Course Learning Outcomes:

Upon successful completion of this course, students will be able to:

1. explain the TCP/IP architecture, IPv4 and IPv6 addressing, NAT, and address resolution mechanisms like ARP, RARP and DHCP.
2. evaluate various unicast and multicast routing protocols (RIP, OSPF, BGP, MPLS, DVMRP, MOSPF) and choose appropriate routing methods
3. explain transport layer protocols, TCP/UDP mechanisms, and congestion control algorithms
4. apply networking principles to design, configure and optimize network performance using TCP/IP protocols and QoS mechanisms in real-world scenarios.

Syllabus

Unit-I: **(15 hours)**

Introduction: TCP/IP Protocol Suite, Addressing. IPv4 Addresses – Classful addressing, classless addressing, special addresses, ARP: Address Mapping, network address translation (NAT), ARP Protocol, DHCP operation; Transport Layer: Transmission Control Protocol-UDP and TCP, TCP- Connection Establishment and Closure (3-way handshake), Flow Control and Congestion Control, Congestion control algorithms- Leaky bucket and Token bucket algorithms.

Unit-II: **(10 hours)**

Quality of Service in TCP/IP networks: Application requirements and performance metrics (bandwidth, delay, jitter, loss), traffic shaping and traffic policing, packet scheduling algorithms – first-in, first-out queuing, priority queuing, fair queuing, weighted fair queuing, admission control, Integrated services and resource reservation protocol (RSVP), differentiated services (Expedited forwarding, assured forwarding)

Unit-III: **(10 hours)**

IP version 6: Introduction, IPv6 features overview, IPv6 header format, extension headers, IPv6 addressing, traffic class, flow labels, IPv6 security, packet sizes, DNS in IPv6- format of IPv6 resource records, DHCP in IPv6 – DHCPv6 messages, Internet Transition: Migrating from IPv4 to IPv6, Dual IP stack implementation-IPv6/IPv4 node, tunnelling, interoperability

Unit-IV: **(10 hours)**
Routing Protocols: static vs. dynamic routing, unicast routing protocols, intra and interdomain routing-Routing Information Protocol (RIP), Open Shortest Path First (OSPF), BGP (Border Gateway Protocol) and MPLS (Multi-Protocol Label Switching); Multicast Routing Protocols: multicasting vs multiple unicasting, MOSPF (Multicast Link State Routing), DVMRP (Multicast Distance Vector).

Readings:

1. Forouzan, Behrouz A. *Data communications and networking*. McGraw Hill, 2012.
2. Comer, Douglas E. *Internetworking with TCP/IP Principles, Protocol, and Architecture.* 6th edition, Pearson Education, 2013.
3. Parziale, Lydia, Wei Liu, Carolyn Matthews, Nicolas Rosselot, Chuck Davis, Jason Forrester, and David T. Britt. T*CP/IP tutorial and technical overview.* (IBM Redbook), 2006 http://www.redbooks.ibm.com/abstracts/gg243376.html
4. Kurose, James, and Keith Ross. *Computer Networking: A Top-Down Approach*. Pearson, 2016.
5. Tanenbaum, Andrew S. *Computer Networks*. Sixth edition. Pearson Education, 2022.

List of Practicals :

1. Find the physical and logical address of your machine by checking your computer's network settings. Further, execute the following computer networking command in Linux/Windows/Mac OS using the command prompt/terminal
1. ipconfig 2. tracert 3. ping 4. netstat 5. nslookup 6. arp 7. route 8. Pathping **(4 hours)**

2. Download ARP trace from here: https://kevincurran.org/com320/labs/wireshark/trace-arp.pcap and open the file downloaded in Wireshark. Set a display filter for packets with the Ethernet address **00:25:64:d5:10:8b.** Find and select an ARP request for the default gateway and examine its fields. **(4 hours)**

3. You are an IT admin for ABC company. You had a report that Ram (a new employee) cannot browse or email with his laptop. After researching, you found that Sita, sitting next to Ram, can browse without any problem. Compare the capture file from both machines and find out why Ram's machine is not online. (Files provided: File: Sita.pcap, Ram.pcap) **(2 hours)**

4. Download the Cisco 3600 Series Router and load the IOS Image in the GNS3 Emulator. Assign IP addresses to the routers and their interfaces. Show the static routing using two routers. **(4 hours)**

5. Design a topology with 2-3 routers and multiple LAN segments (subnets). Configure IPv4 addresses, subnet masks, and default gateways on routers and virtual PCs (VPCS). Further, verify connectivity between hosts in different subnets using ping. Capture traffic using Wireshark to observe source/destination IP addresses and ICMP packets. **(2 hours)**

6. Download the Cisco 7200 Series router and load the IOS Image in the GNS3 emulator. Assign IP addresses to PCs via DHCP configuration **(4 hours)**

7. Implement and observe one-to-one static NAT in the GNS3 Emulator **(2 hours)**

8. Implement unicast routing protocol RIP in GNS3 Emulator. **(2 hours)**

9. Implement the OSPF routing protocol in the GNS3 Emulator and then capture its packets using Wireshark. **(4 hours)**

10. Configure IPv6 global unicast addresses and link-local addresses statically on routers and VPCS. Capture traffic to analyse IPv6 headers. **(2 hours)**

DSE104: Internet of Things [3-0-1]

Course Code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice	
DSE104	**4**	**3**	**0**	**1**	**Knowledge of Computer Network concepts**

Course Objectives:

To introduce the terminology, technology of IoT, and its applications. Introduce the concept of M2M (machine to machine) with necessary protocols, introduce the Python Scripting Language, which is used in many IoT devices, introduce the Raspberry PI platform, which is widely used in IoT applications and implementation of web-based services on IoT devices.

Course Learning Outcomes:

Upon successful completion of this course, the student will be able to

1. understand IoT value chain structure (device, data cloud), application areas, and technologies.
2. understand IoT sensors and the technological challenges IoT devices face, focusing on wireless, energy, power, and sensing modules.
3. market forecast for IoT devices with a focus on sensors.
4. explore and learn about the Internet of Things with the help of preparing projects designed for Raspberry Pi

Syllabus:

Unit-I **(10 Hours)**

Introduction to Internet of Things, definition and characteristics, physical design of IoT, communication models, communication APIs, Wireless Sensor networks, cloud computing, embedded systems, IoT applications: home, smart city, environment, energy, agriculture, and industry

Unit-II **(10 Hours)**

IoT and M2M- software-defined networks, network function virtualization, the difference between SDN and NFV for IoT, basics of IoT system management with NETCONF, YANG-NETCONF, YANG, SNMP NETOPEER

Unit-III **(11 Hours)**

Introduction to Arduino and raspberry Pi- installation, interfaces (serial, SPI, I2C), connecting LED, buzzer, switching high power devices with transistors, controlling AC power devices with Relays, controlling servo motor, speed control of DC Motor, unipolar and bipolar stepper motors

Unit-IV **(14 Hours)**

Sensors- light sensor, temperature sensor with thermistor, voltage sensor, ADC and DAC, temperature and humidity sensor DHT11, motion detection sensors, wireless Bluetooth sensors, level sensors, USB sensors, embedded sensors, distance measurement with ultrasound sensor.

IoT physical servers, cloud storage models, communication APIs web server, web server for IoT, cloud for IoT, python web application framework designing a RESTful web API.

Readings:

1. Bahga Arshdeep, *Internet of Things: A Hands-On Approach*, Bahga & Madissetti, 2014.
2. Misra Sudip, Anandarup Mukherjee, and Arijit Roy, *Introduction to IoT*, Cambridge University Press, 2021.
3. Wallace, Shawn, Matt Richardson, and Wolfram Donat. *Getting started with raspberry pi*. Maker Media, 2021.
4. Monk, S., *Raspberry Pi cookbook: Software and hardware problems and solutions*, O'Reilly Media, 2016.
5. Waher, Peter, *Learning internet of things*, Packt publishing, 2015.

List of Practicals :

1. Introduction to Arduino / NodeMCU / ESP32 in terms of basic GPIO programming. **(2 Hours)**
2. Connect and read data from a PIR sensors or ultrasonic sensors. **(2 Hours)**
3. Read temperature from LM35 or potentiometer using ADC. **(2 Hours)**
4. Wi-Fi Connectivity in NodeMCU / ESP32]. **(2 Hours)**
5. Sending sensor data to the cloud using ThingSpeak or Blynk APIs. **(2 Hours)**
6. Control an LED or relay module via Blynk app or HTTP APIs. **(2 Hours)**
7. Collect and visualize temperature and humidity data using sensors. **(2 Hours)**
8. Publish/subscribe sensor data using MQTT protocol. **(2 Hours)**
9. Build a webpage hosted on the microcontroller to show live sensor data. **(2 Hours)**
10. Automate light/fan control using IR sensor and relay. **(2 Hours)**

11. IoT-based weather monitoring station**. (2 Hours)**
12. Display data on LCD or upload to cloud. **(2 Hours)**
13. Design IoT-based smart parking system. **(2 Hours)**
14. Use ultrasonic sensors to detect parking availability. **(2 Hours)**
15. Read moisture level and send alert via app or email **(2 Hours)**

DSE105: Graph Theory [3-0-1]

Course Code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice	
DSE105	**4**	**3**	**0**	**1**	**Basic understanding of graphs**

Course Objectives:

This course will thoroughly introduce the basic concepts of graph theory, graph properties and formulations of typical graph problems. The student will learn to model diverse applications in many areas of computing, social and natural sciences.

Course Learning Outcomes:

Upon successful completion of this course, the student will be able to

1. model problems using different types of basic graphs like trees, spanning tree, bipartite and planar graphs.
2. understand and identify special graphs like Euler graphs and Hamiltonian graphs.
3. have increased ability to understand various forms of connectedness in a graph.
4. appreciate different graph-coloring problems, matching problems and their solutions.

Syllabus:

Unit-I **(9 Hours)**

Fundamental Concepts: Definitions, examples of problems in graph theory, adjacency and incidence matrices, isomorphisms, paths, walks, cycles, components, cut-edges, cut-vertices, bipartite graphs, Eulerian graphs, Hamiltonian graphs, vertex degrees, reconstruction conjecture, extremal problems, degree sequences, directed graphs, de Bruijn cycles, orientations and tournaments, The Chinese postman problem.

Unit-II **(9 Hours)**

Trees: Trees and forests, characterizations of trees, spanning trees, radius and diameter, enumeration of trees, Cayley's formula, Prüfer code, counting spanning trees, deletion-contraction, the matrix tree theorem, graceful labeling, minimum spanning trees (Kruskal's algorithm), shortest paths (Dijkstra's algorithm).

Unit-III **(16 Hours)**

Matching and Covers: Matchings, maximal and maximum matchings, M-augmenting paths, Hall's theorem and consequences, Min-max theorems, maximum matchings and vertex covers, independent sets and edge covers, Connectivity, vertex cuts, Edge-connectivity.

Connectivity and Paths: Blocks, k-connected graphs, Menger's theorem, line graphs, dual graphs, network flow problems, flows and source/sink cuts, Ford-Fulkerson algorithm, Max-flow min-cut theorem.

Unit-IV **(11 Hours)**

Graph Coloring: Vertex colorings, bounds on chromatic numbers, Chromatic numbers of graphs constructed from smaller graphs, chromatic polynomials, properties of the chromatic polynomial, the deletion-contraction recurrence.

Planar Graphs: Planar graphs, Euler's formula, Kuratowski's theorem, five- and four-color theorems.

Readings:

1. West, Douglas B, *Introduction to Graph Theory*, 2^{nd} Edition, Pearson, 2017.
2. Chartrand, Gary and Zhang, Ping, *Introduction to Graph Theory*, Tata McGraw Hill, 2017.
3. Gross, Jonathan L., Yellen, Jay and Anderson, Mark, *Graph Theory and Its Applications*, 3^{rd} Edition, Taylor & Francis, 2024.

References:

1. Deo, Narsingh, *Graph Theory with Applications to Engineering and Computer Science*, Prentice Hall India Learning Private Limited, New edition, 1979.

List of practicals:

You may choose a suitable programming language and all necessary/relevant inputs/ problems.

1. Write a program to check whether the graph is: Bipartite, Eulerian, Hamiltonian or not. **(4 Hours)**

2. Write a program to find the minimum spanning trees of the graph. **(2 Hours)**

3. Write a program to find the shortest paths of the graph. **(2 Hours)**

4. Write a program to implement the Prüfer code of the graph. **(2 Hours)**

5. Write a program to implement the maximum and maximal matching of the graph. **(4 Hours)**

6. Write a program to implement the Ford-Fulkerson algorithm, Max-flow, and Min-cut theorem of the graph. **(6 Hours)**

7. Write a program to implement the Vertex colouring of the graph. **(2 Hours)**

8. Write a program to find the chromatic numbers of the graph. **(2 Hours)**

9. Write a program to check whether the graph is planar or not. **(2 Hours)**

10. Write a program to solve the Chinese postman problem. **(4 Hours)**

DSE106: Soft Computing [3-0-1]

Course Code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/Practice	
DSE106	**4**	**3**	**0**	**1**	**Graduation**

Course Objectives:

The objective of the course is to understand and apply different domains of soft computing techniques like neural networks, fuzzy logic, genetic algorithm and swarm optimization. This course provides insights of soft computing frameworks applicable to bring its precision solutions for wide range of complex scientific applications.

Course Learning Outcomes:

Upon successful completion of this course, students will be able to

1. understand different soft computing techniques.
2. design hybrid soft techniques over conventional computing methods.
3. design robust and low-cost intelligent machines to solve real-world problems.

Syllabus:

Unit-I **(16 Hours)**

Fundamental Concepts: Introduction of soft computing, soft computing vs. hard computing, various types of soft computing techniques, applications of soft computing.

Artificial Neural Networks and Paradigms: Introduction to neuron model, neural network architecture, learning rules, perceptrons, single layer perceptrons, multilayer perceptrons, back propagation networks, Kohonen's Self organizing networks, Hopfield network, applications of neural networks.

Unit-II **(9 Hours)**

Fuzzy Logic: Introduction of fuzzy sets and fuzzy reasoning, basic functions on fuzzy sets, relations, rule-based models and linguistic variables, fuzzy controls, fuzzy decision making, inferencing, defuzzification, fuzzy clustering, fuzzy rule-based classifier, applications of fuzzy logics.

Unit-III **(12 Hours)**

Evolutionary Algorithms: Introduction to evolutionary algorithms, basic principles of evolutionary algorithms, evolutionary strategies, genetic algorithm, fitness computations, cross-over, mutation, evolutionary programming, classifier systems, genetic programming parse trees, variants of genetic algorithm, genetic algorithm applications.

Unit-IV **(8 hours)**

Swarm Optimizations: Ant Colony Optimization, Particle Swarm Optimization, Artificial Bee Colony Optimization, concept of multi-objective optimization problems (MOOPs), Multi-Objective Evolutionary Algorithm (MOEA), Non-Pareto approaches to solve MOOPs, Pareto-based approaches to solve MOOPs, Some applications with MOEAs.

Readings:

1. Jang, Jang-Shing Roger, Sun, Chuen-Tsai, and Eiji, Mizutani, *Neuro-Fuzzy and Soft Computing*: A *Computational Approach to Learning and Machine Intelligence*, 1st edition, Pearson Education, 2015.
2. Haykin, Simon O., *Neural Networks: A Comprehensive Foundation*, 2nd edition, Pearson, 1998.
3. Zimmermann, Hans-Jurgen, *Fuzzy Set Theory - and its Applications*, 2nd edition, Kluwer Academic Publishers, 1991.
4. Goldberg, David E., *Genetic Algorithms in Search, Optimization, and Machine Learning* Addison Wesley, 1989.

References:

1. Yegnanarayana, B., Artificial Neural Networks, PHI, 2004.
2. Zurada, Jacek M., Introduction to Artificial Neural Systems, 1st edition, Jaico Publishing House, 1994.
3. Ross, Timothy J., Fuzzy Logic with Engineering Applications, 3rd edition, Wiley, 2011

List of practicals:

You may choose a suitable programming language and all necessary/relevant inputs/ problems.

1. Write a program to implement single-layer and multi-layer neural networks. **(2 Hours)**

2. Write a program to implement back-propagation neural networks. **(2 Hours)**

3. Write a program to implement Kohonen's self-organising and Hopfield networks. **(3 Hours)**

4. Write a program to implement fuzzy: membership functions, controller systems, defuzzification, and rules. **(3 Hours)**

5. Write a program to implement/design the working of (i) Anti-lock Braking System and (ii) FAN speed control of Room Fan using a fuzzy system. **(6 Hours)**

6. Write a program for $Max : f(x) = x_1^3 - 2x_2^2 + 3x_3^3 + 4x_4$ that use Genetic Algorithms to solve the problem of Maximization function. **Subject to: $1 \leq x_1 \leq 5$; $2 \leq x_2 \leq 4$; $3 \leq x_3 \leq 6$; $1 \leq x_4 \leq 10$**, Genetic Algorithm will be used to find the value of x_1, x_2, x_3, x_4 and $f(x)$ that satisfy the above equation. **(2 Hours)**

7. Write a program to implement the Ant Colony Optimization technique. **(3 Hours)**

8. Write a program to implement the Particle Swarm Optimization technique. **(3 Hours)**

9. Write a program to implement the Artificial Bee Colony Optimization technique. **(3 Hours)**

10. Write a program to implement the Multi-objective evolutionary algorithm. **(3 Hours)**

List of GEs for Semester I

GE101: DATA ANALYSIS AND VISUALIZATION [3-0-1]

Course Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
GE101	**4**	**3**	**0**	**1**		**Basic understanding of statistics and familiarity with Python programming**

Course Objectives:

The course develops students' competence in cleaning and analyzing data related to a chosen application. It also aims to develop skills in using various tools for data visualization and choosing the right tool for given data.

Course Learning Outcomes:

On completing the course, students will be able to:

1. use data analysis tools with ease.
2. load, clean, transform, merge, and reshape data.
3. create informative visualisations and summarise data sets.
4. analyse and manipulate time series data.
5. solve real-world data analysis problems.

Syllabus

Unit-I **(10 hours)**

Introduction: Introduction to Data Science, Exploratory Data Analysis and Data Science Process. Motivation for using Python for Data Analysis, Introduction of Python shell iPython and Jupyter Notebook. Essential Python Libraries: NumPy, pandas, matplotlib, SciPy, scikit-learn, stats models.

Unit-II **(10 hours)**

Getting Started with Pandas: Arrays and vectorized computation, Introduction to pandas Data Structures, Essential Functionality, Summarizing and Computing Descriptive Statistics. Data Loading, Storage and File Formats. Reading and Writing Data in Text Format, Web Scraping, Binary Data Formats, Interacting with Web APIs, Interacting with Databases, Data Cleaning and Preparation. Handling Missing Data, Data Transformation, String Manipulation

Unit-III **(15 hours)**

Data Wrangling: Hierarchical Indexing, Combining and Merging Data Sets Reshaping and Pivoting. Data Visualization Matplotlib: Basics of Matplotlib, plotting with pandas and seaborn, and other Python visualization tools. Data Aggregation and Group operations: Data grouping, Data aggregation, General split-apply-combine, Pivot tables and cross-tabulation.

Unit-IV **(10 hours)**

Time Series Data Analysis: Date and Time Data Types and Tools, Time series Basics, Frequencies and Shifting, Time Zone Handling, Periods and Periods Arithmetic, Resampling and Frequency Conversion, Moving Window Functions.

Readings

1. McKinney, Wes. *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython*. " O'Reilly Media, Inc.", 2012.
2. O'Neil, Cathy, and Rachel Schutt. *Doing data science: Straight talk from the frontline*. " O'Reilly Media, Inc.", 2013.

List of Practicals :

1. Install and explore Jupyter Notebook and iPython shell. Write Python code snippets using basic data types, control structures, and functions. Import and demonstrate the use of libraries like NumPy, pandas, matplotlib, and scikit-learn with simple examples. **(4 hours)**
2. Perform array creation, indexing, slicing, broadcasting, and vectorized operations using NumPy. Apply mathematical functions, linear algebra routines, and random number generation. **(4 hours)**
3. Create and manipulate Series and DataFrames. Demonstrate operations like indexing, filtering, sorting, reindexing, and applying functions across rows/columns. Use pandas for basic descriptive statistics and summaries. **(4 hours)**
4. Read data from CSV, Excel, JSON, and HTML files using pandas. Demonstrate web scraping using requests and BeautifulSoup. Interact with Web APIs and connect to SQLite databases. Save and load data in binary formats (e.g., pickle, HDF5). **(4 hours)**
5. Handle missing values, duplicate data, and inconsistent formats. Apply string operations and regular expressions for text data cleaning. Demonstrate data transformations such as normalization, binning, and type conversion**. (2 hours)**
6. Merge and join multiple DataFrames. Apply hierarchical indexing and reshaping techniques like stack, unstack, melt, and pivot. Create pivot tables and perform cross-tabulations. **(2**

hours)

7. Use groupby() to aggregate and summarize data. Demonstrate split-apply-combine strategy. Compute custom aggregation functions and perform multi-level group operations. **(2 hours)**
8. Create line, bar, histogram, and scatter plots using matplotlib. Use pandas and seaborn for advanced plots like boxplots, pair plots, heatmaps, and violin plots. Customize plots with titles, labels, legends, and annotations. **(4 hours)**
9. Work with date and time objects in pandas. Parse date strings, index data with datetime, and perform slicing and selection. Resample data to different frequencies and demonstrate upsampling/downsampling. **(2 hours)**
10. Apply moving window functions (rolling, expanding). Handle time zones and perform shifting, lagging, and differencing. Create and manipulate period objects and apply arithmetic on time series data. **(2 hours)**

GE102: PROGRAMMING WITH PYTHON [3-0-1]

Course Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
GE 102	**4**	**3**	**0**	**1**		**Graduation**

Course Objectives:

The course aims to develop the student's problem-solving skills. The course also focuses on debugging skills. The student learns to develop well-documented modular wode.

Course Learning Outcomes:

At the end of the course, students will be able to

1. select a suitable programming construct and in-built data structure for a given problem.
2. design, develop, document, and debug modular programs.
3. use recursion as a programming paradigm for problem-solving.
4. apply object-oriented paradigm for problem-solving.

Syllabus:

Unit-I **(9 hours)**

Introduction: Notion of class, object, identifier, keyword, and literal; basic data types: int, float, string, Boolean; basic operators (arithmetic, relational, logical, assignment), standard libraries.

Unit-II **(10 hours)**

Program Development: Modular program development, input and output statements, control statements: branching, looping, exit function, break, continue, and switch-break; use of mutable and immutable structures. strings, lists, sets, tuples and dictionary, and associated operations testing, and debugging a program.

Unit-III **(18 hours)**

Recursion: Use of recursion as a programming paradigm for problem solving.
Object Oriented Programming: Use of classes, inheritance, and operator overloading in problem solving. Exception Handling and File Handling: Reading and writing text and structured files, errors and exceptions.

Unit-IV **(8 hours)**
Visualization using 2D and 3D graphics: Visualization using graphical objects like point, line, histogram, 3D objects, animation.

Readings:

1. Allen Downey, *Think Python: How to Think Like a Computer Scientist*, 3rd edition, O'REILLY Publishers, 2024.
2. J.V. Guttag, *Introduction to Computation and Programming Using Python: With Application to Understanding Data*, MIT Press, 2016.
3. Robert Sedgewick , Kevin Wayne, Robert Dondero, *Introduction to Programming in Python: An Interdisciplinary Approach, Addison-Wesley Professional, 2015*
4. Tony Gaddis, *Starting Out with Python*, Pearson, 2021

List of Practicals :

1. Write a program that accepts height and weight from the user and calculates the Body Mass Index (BMI), displaying the BMI value and its category (underweight, normal, overweight, or obese). Use appropriate data types and arithmetic operators. **(2 hours)**
2. Write a program that takes a list of numbers as input from the user and provides options to find the maximum, minimum, average, and sorted version of the list using built-in list functions. **(2 hours)**
3. Write a menu-driven program using modular functions that performs the following operations on sets: union, intersection, difference, and symmetric difference. Allow users to input two sets of integers. **(2 hours)**
4. Develop a program that simulates a basic dictionary-based glossary. Allow the user to add, update, delete, and retrieve definitions of terms using a dictionary data structure. **(2 hours)**

5. Write a menu-driven program that accepts a string input from the user and provides the following options: (a) check whether the given string is a palindrome, and (b) count the number of vowels and consonants in the string. The program should continue to display the menu until the user chooses to exit. Ensure that the implementation handles both uppercase and lowercase characters and ignores spaces and punctuation where appropriate. **(2 hours)**

6. Write a recursive program for the following: **(2 hours)**

   - To compute the factorial of a given non-negative integer.
   - To find n^{th} number in the Fibonacci series
   - To add multiplies two number without using multiplication operator.

- To count number of characters in a string
- To add items of a list

7. Write a program to implement a Student Record System using object-oriented programming concepts. Create a class called Student that includes attributes such as student ID, name, age, department, and marks. Define methods to input student details, store them in an appropriate data structure and display the details of a student based on their student ID. The program should allow the user to add multiple students and retrieve information of any specific student by searching with the ID. Use constructors for initializing student objects and demonstrate encapsulation by keeping attributes private and accessing them through getter methods. **(2 hours)**
8. Write a program to implement a Bank Account Management System using object-oriented programming principles. Begin by creating a base class called BankAccount that includes attributes such as account number, account holder name, and balance. Define member functions to perform common operations like deposit(amount), withdraw(amount), and check_balance() for displaying the current balance. **(2 hours)**
9. Write a program to implement a Library Management System. Create classes Book, User, and BorrowedBook. Use inheritance and method overriding for different user types (Student, Faculty) with borrowing limits. Include features to borrow and return books. **(2 hours)**
10. Write a program that reads a text file containing names and marks of students, calculates average marks, and writes the result into another file. Handle exceptions for file not found, read/write errors, and invalid data. **(2 hours)**
11. Develop a contact management system that stores contacts in a structured text file (CSV format). Implement the functionalities to add, update, delete, and search contacts. Use try-except blocks to handle runtime errors like missing fields or invalid input. **(2 hours)**
12. Write a program using a graphics library (e.g., matplotlib or turtle) to draw basic 2D shapes such as points, lines, rectangles, and circles based on user input coordinates and dimensions. **(2 hours)**
13. Write a program that visualizes data using a histogram. Accept a set of numbers (e.g., student marks) and plot the distribution of scores using matplotlib. Label axes and add a title. **(2 hours)**
14. Write a program that plots a 3D surface or wireframe using matplotlib or a suitable 3D plotting library. Allow the user to input a mathematical function and visualize its graph. **(2 hours)**
15. Create a data visualization dashboard that takes user-defined numeric data and shows a bar chart, line graph, and pie chart on demand. Allow toggling between different chart types. **(2 hours)**

SEMESTER - II

DSC201: INFORMATION SECURITY [3-0-1]

Course Code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice	
DSC201	**4**	**3**	**0**	**1**	**Graduation**

Course Objectives:

This course aims to provide a comprehensive understanding of security principles, threats, and cryptographic techniques used in modern computing systems. It covers security models, authentication mechanisms, and key management protocols to safeguard data and communication. The course also emphasizes real-world applications of security frameworks, to enhance secure communication and data protection.

Course Learning Outcomes:

Upon successful completion of this course, students will be able to:

1. analyze the distinction between system protection and security, emphasizing their role in safeguarding digital assets
2. implement symmetric and asymmetric encryption algorithms
3. describe the role and implementation of digital signatures.
4. understand how cryptographic methods and security models are applied in practical systems like secure communication and e-commerce

Syllabus:

Unit-I **(4 hours)**

Overview of Security: Protection versus security; aspects of security– confidentiality, data integrity, availability, privacy; user authentication, access controls. Security Threats and Models: Program threats, worms, viruses, Trojan horse, trap door, stack and buffer overflow; system threats- intruders; communication threats- tapping and piracy.

Unit-II **(20 hours)**

Cryptography: Substitution, transposition ciphers, symmetric-key algorithms: Data Encryption Standard, Advanced Encryption Standard, IDEA, block cipher operation, stream ciphers: RC-4. Public key encryption: knapsack, RSA, El Gamal, Elliptic curve cryptography.

Unit-III **(15 hours)**

Integrity and authentication: Message Integrity - MDC, MAC, cryptographic hash function - iterated hash functions, compression functions, SHA-512, Digital signatures -RSA digital signature scheme, ElGamal digital signature scheme, digital signature standard (DSS), elliptic curve digital signature scheme, Entity authentication.

Unit-IV **(6 hours)**

Key Management and Intrusion Detection & Prevention: Symmetric key distribution, Kerberos, Diffie-Hellman key agreement, public key distribution, public key infrastructures, Intrusion Detection and Prevention Systems.

Readings:

1. Stalling, William. *Cryptography and Network Security Principles and Practices*. 8th edition, Pearson education, 2023
2. Forouzan, Behrouz A., and Debdeep Mukhopadhyay. *Cryptography and Network Security.* 3rd edition. McGraw-Hill education, 2015.
3. Elbirt, Adam. *J Understanding and Applying Cryptography and Data Security*. CRC Press, Taylor Francis Group, 2015.
4. Pfleeger, Charles P. SL Pfleeger, Jonathan Margulies. *Security in Computing*. 5th edition. Prentice-Hall of India, 2018.

List of practicals :

Students may choose a suitable programming language to do the following lab exercises.

1. Implement the extended Euclidean algorithm **(2 hours)**
2. Implement Encryption and decryption using the Additive Cipher **(2 hours)**
3. Cryptanalysis of the Additive Cipher **(2 hours)**
4. Implement Encryption and decryption using the Multiplicative Cipher **(2 hours)**
5. Cryptanalysis of the Multiplicative Cipher **(2 hours)**
6. Implementation of columnar transposition cipher for encryption and decryption of messages. **(2 hours)**
7. Encryption and decryption of files using python's cryptography library to demonstrate symmetric encryption and asymmetric encryption algorithms discussed in the class **(10 hours)**
8. Computation of SHA-512 Hash and HMAC for file integrity verification using a Shared Secret Key. **(4 hours)**
9. Verification of file integrity and displaying "Integrity Verified" or "Integrity Compromised" Based on various hashing algorithms. **(4 hours)**

DSC202: DEEP LEARNING [3-0-1]

Course Code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/Practice	
DSC202	**4**	**3**	**0**	**1**	**Graduation**

Course Objectives:
The student learns various state-of-the-art deep learning algorithms and their applications to solve real-world problems. The student develops skills to design neural network architectures and training procedures using various deep learning platforms and software libraries.

Course Learning Outcomes:
On completing this course, students will be able to:

1. describe the feedforward and deep networks.
2. design single and multi-layer feed-forward deep networks and tune various hyper-parameters.
3. analyze the performance of deep networks.

Syllabus:

Unit-I **(14 hours)**
Introduction: Historical context and motivation for deep learning; deep feedforward neural networks, regularizing a deep network, model exploration, and hyperparameter tuning. Convolution Neural Networks: Introduction to convolution neural networks: stacking, striding, and pooling, applications like image and text classification.

Unit-II **(17 hours)**
Introduction to Natural Language Processing (NLP), Traditional NLP Techniques, Sequence Modeling: Recurrent Nets: Unfolding computational graphs, recurrent neural networks (RNNs), bidirectional RNNs, encoder-decoder sequence to sequence architectures, deep recurrent networks. Large Language Models: Transformer Architecture, Pre-training and Fine-tuning Language Models, Ethical Considerations and Bias in Language Models, Applications of Large Language Models (Text Generation, Sentiment Analysis, Question Answering)

Unit-III **(10 hours)**
Autoencoders: Undercomplete autoencoders, regularized autoencoders, sparse autoencoders, denoising autoencoders, representational power, layer, size, and depth of autoencoders, stochastic encoders and decoders. Generative Adversarial Networks (GANs): Introduction to Generative Adversarial Networks, GAN Architectures (DCGAN, CycleGAN), Applications of GANs (Image Generation, Style Transfer)

Unit-IV **(4 hours)**
Structuring Machine Learning Projects: Orthogonalization, evaluation metrics, train/dev/test distributions, size of the dev and test sets, cleaning up incorrectly labelled data, bias and variance with mismatched data distributions, transfer learning, multi-task learning.

Readings:

1. Ian Goodfellow, *Deep Learning,* MIT Press, 2016.
2. Jeff Heaton, *Deep Learning and Neural Networks*, Heaton Research Inc, 2015.
3. Mindy L Hall, *Deep Learning*, VDM Verlag, 2011.
4. Li Deng (Author), Dong Yu, *Deep Learning: Methods and Applications (Foundations and Trends in Signal Processing),* Now Publishers Inc, 2009.

List of Practicals :

1. Implement a deep neural network using PyTorch or TensorFlow on a simple dataset. Apply techniques for regularization, such as dropout and early stopping. Tune hyperparameters like learning rate, number of layers, and batch size. **(4 hours)**
2. Design and train a convolutional neural network on an image dataset. Apply stacking, striding, pooling, and normalization techniques. Evaluate model performance using accuracy and confusion matrix. **(4 hours)**
3. Preprocess text data, tokenize and vectorize inputs, and implement a CNN-based model for sentiment classification. **(2 hours)**
4. Implement a basic RNN to model sequences, such as character-level language modeling. Visualize unfolding of the computational graph and train the model on a small text corpus. **(4 hours)**
5. Build a sequence-to-sequence model using an encoder-decoder architecture with bidirectional RNNs for tasks like machine translation. Evaluate output quality using BLEU score or accuracy. **(2 hours)**
6. Use Transformers to load a pre-trained BERT or GPT model. Fine-tune it on a downstream task such as sentiment analysis or question answering datasets. **(4 hours)**
7. Analyze generated outputs from a large language model for biased or unethical responses. **(2 hours)**
8. Implement different types of autoencoders (undercomplete, sparse, denoising) on the MNIST dataset. Visualize reconstructed images and evaluate reconstruction error. **(4 hours)**
9. Build and train a simple GAN on MNIST or CelebA dataset. Generate synthetic images and compare visual quality with real images. Track generator and discriminator loss over time. (**2 hours)**
10. Use transfer learning with a pre-trained CNN (e.g., ResNet) to classify a small custom dataset. Discuss best practices for train/dev/test splits, evaluation metrics, and handling label noise. Experiment with multi-task learning if time permits. (**2 hours)**

DSC203: Wireless and Mobile Communications [3-0-1]

Course Code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/Practice	
DSC203	**4**	**3**	**0**	**1**	**Graduation**

Course Objectives:

This course aims to provide a comprehensive understanding of wireless communication principles, including propagation mechanisms, fading effects, and multiple access techniques. It explores IEEE 802.11 WLANs and cellular communication concepts such as frequency reuse, handoff strategies, and interference management. It provides insights into ad-hoc and sensor networks, regulatory aspects, and network design challenges, preparing students to apply these concepts in real-world scenarios and emerging wireless technologies.

Course Learning Outcomes:

On completion of the course, students will be able to:

1. analyze and simulate wireless communication scenarios to observe propagation effects, fading impacts, and the performance of multiple access techniques.
2. understand and evaluate cellular network design by analyzing frequency reuse, handoff strategies, interference management, and capacity enhancement techniques
3. compare GSM, CDMA, LTE, WiFi, and WiMAX while understanding regulatory and spectrum management
4. analyze the architecture and key factors influencing the deployment of wireless sensor networks (WSNs) and vehicular ad-hoc networks (VANETs)

Syllabus:

Unit-I **(10 hours)**

Introduction: Wireless communication systems, history and evolution (1G to 5G and beyond), Radio wave propagation mechanisms (reflection, diffraction, scattering), large-scale path loss models, small-scale fading and multipath propagation, doppler effect and delay spread, spread spectrum techniques, multiple access and duplexing techniques

Unit-II **(10 hours)**

Cellular mobile communication: frequency re-use and channel assignment strategies, handoff strategies, types, priorities, practical considerations, interference and system capacity, co-channel and adjacent channel interference, power control measures, grade of service, definition, standards, coverage and capacity enhancement in cellular network, cell splitting, sectoring, microcells

Unit-III **(10 hours)**

Wireless systems and standards: Global System for Mobile (GSM) - services and features, system architecture, radio sub-system, channel types (traffic and control), frame structure, signal processing, CDMA standards-frequency and channel specifications, Forward and Reverse CDMA channels, WiFi, WiMAX, UMB, UMTS, LTE, and recent trends, Regulatory issues (spectrum allocation, spectrum pricing, licensing, tariff regulation and interconnection issues)

Unit-IV **(15 hours)**

IEEE 802.11(Wireless LAN) Standards: IEEE 802.11 standards overview (802.11a/b/g/n), physical layer, medium access control layer-CSMA/CA mechanism, RTS/CTS mechanism for collision avoidance, hidden node and exposed node problems, QoS enhancements in IEEE 802.11, future trends in IEEE 802.11, IEEE 802.11 and IEEE 802.3 (Ethernet) interoperability, IEEE 802.11 in cellular networks (5G and Wi-Fi Offloading); Ad-hoc networks and sensor networks: introduction, challenges and issues, AODV, DSR, DSDV routing protocols; architecture and factors influencing the sensor network design; concept of MANET and VANET

Readings:

1. Rappaport, Theodore S. *Wireless communications: principles and practice*. Cambridge University Press, 2024.
2. Murthy, C. Siva Ram, and B. S. Manoj. *Ad hoc wireless networks: Architectures and protocols*. Pearson education, 2008.
3. Stallings, William. *Wireless communications & networks*. Pearson Education India, 2009.
4. Goldsmith, Andrea. *Wireless communications.* Cambridge university press, 2013.
5. Garg, Vijay. *Wireless communications & networking*. Elsevier, 2010.
6. Carlos,M.d. and Agrawal, D P. *Ad Hoc and Sensor Networks: Theory and Applications*, World scientific publishing company, 2011.

List of Practicals

1. Write a program to: **(4 hours)**
   a) determine the free-space path loss and the power received.
   b) endorse the statement: “Free Space Attenuation increases by 6 dB whenever the length of the path is doubled. Similarly, as frequency is doubled, free space attenuation also increases by 6 dB”.
   c) plot a chart between PL (dB) and distance (Km) with varying frequencies.

2. Write a program to: **(4 hours)**
   (a) determine the two-ray path loss and the power received.
   (b) Endorse the statement: "The total attenuation increases by 12 dB when the separation distance is doubled” in the two-ray model.
   (c) plot a chart between PL(dB) and distance (Km) with varying frequencies for two ray path loss model.
   (d) Further, plot a chart between PL(dB) and distance (Km) comparing the free space path loss model and two-ray model with varying frequencies.

3. Write a program to generate a PN (Pseudo-Noise) sequence using a Linear Feedback Shift Register (LFSR). **(2 hours)**

4. Write a program to: **(4 hours)**
   a) Calculate the received power at a distance d using a log-normal shadowing model for

both indoor and outdoor environments.

b) plot a graph for Log normal shadowing model between distance and path loss/received power

   i. varying path loss exponent
   ii. varying standard deviation of shadowing parameters.

c) Compare friss-space path loss model and log normal shadowing model

5. Implement a function to create a hexagonal grid with a specified cluster size and cell size. Visualise the hexagonal grid using matplotlib. **(2 hours)**

6. Simulate a cellular network with a given number of base stations (BS) and mobile users randomly distributed in the hexagonal cells. Assign frequencies to each base station. Implement a function to calculate the signal strength at each mobile user's location based on the distance from the serving base station. Consider the free space path loss model for signal propagation. **(4 hours)**

(a)Plot a heatmap or contour plot of signal strength across the cellular network.

(b) Observe how signal strength varies within and between cells.

7. Capture WiFiTraffic on your device using the tool Wireshark. Analyse the interframe gaps for DIFS, SIFS and EIFS. Further, for each data frame, identify the corresponding ACK frame. Check whether the ACK frame is immediately following the data frame and has the SIFS timing gap. If not, determine if there is any delay and discuss possible reasons. **(2 hours)**

8. Write a Python script to simulate cell splitting. Divide some existing cells into smaller cells to demonstrate how this process increases system capacity. **(2 hours)**

9. Simulate a wireless network with three nodes (A, B, C) where A and C are hidden terminals. Implement the RTS/CTS protocol to prevent collisions at B when both A and C attempt to send data simultaneously. **(4 hours)**

10. Capture Wi-Fi frames using Wireshark on your machine’s Wi-Fi interface and observe basic WLAN information. Filter for beacon frames and identify the BSSID and SSID, then find and analyze at least one Data frame and one ACK frame. Note the packet’s source and destination MAC addresses. **(2 hours)**

SBC201: Ethics for Responsible AI [1 -1-0]

Course title	Credits	Credit distribution of the course			Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice	

SBC201	**2**	**1**	**0**	**1**	**Knowledge of AI and ML concepts**

Course Objectives:

The objective of this course is to equip students with foundational concepts of ethics and relate them to the ethical concerns in the rapidly advancing field of Artificial Intelligence. The course also imparts practical skills to the students to address ethical challenges in AI development and deployment through important case studies. Taught using recent research publications, the course aims to develop competence to adapt to new ethical challenges that arise due to rapid advancements in AI.

Course Learning Outcomes:

At the end of this course, students will be able to

1. explain the fundamental theories of ethics.
2. relate the theories to the ethical concerns regarding Artificial Intelligence.
3. understand the practical implications of the ethical concerns.
4. apply mitigation strategies using available tools.
5. equipped to understand and handle new ethical challenges that arise due to swiftly advancing AI technology.

Syllabus

Unit-I **(10 hours)**

Foundations of Ethics & Ethical AI: Introduction to Ethics; Ethical theories - Utilitarianism, Deontology, Virtue Ethics; Case Study - The Trolley Problem in autonomous vehicles; AI-Specific Ethical Challenges - Bias and Fairness, Transparency and Explainability, Accountability, Privacy. Case studies for Ethical concerns and Mitigation: Bias and Fairness, Transparency and Explainability, Accountability, Privacy (hiring, criminal justice, healthcare, autonomous driving etc.), Technical challenges for addressing ethical concerns; Mitigation strategies and techniques - Federated learning, differential privacy, Algorithmic manipulation, Data Curation; Explainability tools (SHAP and LIME).

Unit-II **(5 hours)**

AI Governance: Societal risks - Job displacement, misinformation, environmental costs, deepfake; IEEE guidelines; ACM Code of Ethics for development and deployment of AI; Introduction to "right to explanation" in EU's GDPR, EU AI Act, Indian scenario; Future challenges - AGI Ethics, autonomous weapons.

Reading:

1. Jobin, A., Ienca, M. & Vayena, E. (2019). *The global landscape of AI ethics guidelines.* Nat Mach Intell 1, 389–399, 2019.

2. Radanliev, P., Santos, O., Brandon-Jones, A., & Joinson, A., *Ethics and responsible AI deployment*. Frontiers in Artificial Intelligence, 7, 1377011, 2024.

3. Pant, A., Hoda, R., Spiegler, S. V., Tantithamthavorn, C., & Turhan, B., *Ethics in the age of*

AI: An analysis of AI practitioners' awareness and challenges. ACM Transactions on Software Engineering and Methodology, 33(3), 1-35, 2024

4. Willem, T., Fritzsche, M. C., Zimmermann, B. M., Sierawska, A., Breuer, S., Braun, M., ... & Buyx, A,. *Embedded Ethics in Practice: A Toolbox for Integrating the Analysis of Ethical and Social Issues into Healthcare AI Research*. Science and Engineering Ethics, 31(1), 1-22, 2025.

List of Practicals

1. AI Ethics Case Study Debate (https://www.aiethicist.org/ethics-cases-registries)

Choose an AI ethics case study from the above mentioned URL (e.g., facial recognition in policing, autonomous vehicle accidents, ChatGPT and misinformation). Study the selected case in depth and work in a team of six students.

Each team will role-play the following three stakeholder groups: Developers, Victims of bias and Regulators (Two students per stakeholder group). Prepare 5-minute arguments both in favour of and against the deployment of the AI system under discussion. Each team must present a balanced and well-researched perspective, considering ethical, technical, and societal implications. Next, prepare a one-page individual write-up that should reflect on one of the debates conducted in class, in addition to the one in which they participated. **(14 hours)**

2. Select and download a classification dataset of your choice. Train a complex model, such as a Random Forest or a Neural Network, on the dataset. Use LIME to explain the model's predictions for three correctly classified and three misclassified test instances.

Submit the notebook containing the complete code for data preprocessing, model training, evaluation, and LIME-based explanation. Also, submit a report summarising your findings, including visualisations from LIME and your interpretation of the explanations. **(8 hours)**

3. Choose one of the following datasets.

- MBIC – A Media Bias Annotation Dataset**:** https://arxiv.org/pdf/2105.11910
- COMPAS - Recidivism Racial Bias: https://www.kaggle.com/datasets/danofer/compass
- Adult Income dataset: https://archive.ics.uci.edu/dataset/2/adult
- Dataset provided in the class

Train a classification model (e.g., Decision Tree, Logistic Regression, or XGBoost) on a suitable dataset. Use SHAP (SHapley Additive exPlanations) to interpret the model's predictions. Select five correct and five incorrect predictions and analyse them using SHAP values to understand the key features contributing to each decision. Interpret feature importance and identify patterns that influence the model's decisions. Detect any potential biases or discriminatory behaviour in the model's predictions. Summarise your findings, and reflect on how the identified biases can be mitigated to make the model fairer and more robust. **(8 hours)**

List of DSEs for Semester II

DSE201: Social Networks Analysis [3-0-1]

Course Code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice	
DSE201	**4**	**3**	**0**	**1**	**Basic knowledge of graphs**

Course Objectives:

The course aims to equip students with various SNA approaches to data collection, cleaning, and pre-processing of network data.

Course Learning Outcomes:

On completing this course, students will be able to:

1. explain the basic concepts and principles of social network.
2. identify different types of social networks and their characteristics.
3. implement and apply various social network analysis techniques, such as, influence maximization, community detection, link prediction, and information diffusion.
4. apply network models to understand phenomena such as social influence, diffusion of innovations, and community formation.

<u>Syllabus:</u>

Unit-I **(9 hours)**

Introduction to Social Network Analysis: Introduction to Social Network Analysis, Types of Networks, Nodes Edges, Node Centrality, betweenness, closeness, eigenvector centrality, network centralization, Assortativity, Transitivity, Reciprocity, Similarity, Degeneracy and Network Measure, Networks Structures, Network Visualization, Tie Strength, Trust, Understanding Structure Through User Attributes and Behavior.

Unit-II **(12 hours)**

Link Analysis and Link Prediction: Applications of Link Analysis, Signed Networks, Strong and Weak Ties, Link Analysis and Algorithms, Page Rank, Personalized PageRank, DivRank, SimRank, PathSim. Temporal Changes in a Network, Evaluation Link Prediction Algorithms, Heuristic Models, Probabilistic Models, Applications of Link Prediction.

Unit-III **(12 hours)**

Community Detection: Applications of Community Detection, Types of Communities, Community Detection Algorithms, Disjoint Community Detection, Overlapping Community Detection, Local Community Detection, Evaluation of Community Detection Algorithms.

Unit-IV **(12 hours)**

Influence Maximization: Applications of Influence Maximization, Diffusion Models, Independent Cascade Model, Linear Threshold Model, Triggering Model, Time-Aware Diffusion Model, Non-Progressive Diffusion Model. Influence Maximization Algorithms, Simulation-Based Algorithms, Proxy-Based Algorithms, Sketch-Based Algorithms, Community-Based Influence Maximization, and Context-Aware Influence Maximization.

<u>Readings:</u>

1. Chakraborty, T. *Social Network Analysis*, Wiley India, 2021.
2. Knoke, D. and Yang, S. *Social network analysis*. SAGE publications, 2019.
3. Golbeck, J. *Analyzing the social web*, Morgan Kaufmann, 2013.
4. Wasserman, S and Faust, K. *Social network analysis: Methods and applications*, Cambridge University Press, 2012.
5. Newman, M.E.J. *Networks: An introduction*. Oxford University Press, 2010.
6. Chen, W. Castillo, C and Lakshmanan, L.V.S. *Information and influence propagation in social networks,* Springer Nature, 2014
7. Srinivas, V and Mitra, P. *Link prediction in social networks: role of power law distribution,* New York: Springer International Publishing, 2016

List of Practicals :

1. Implement heuristic link prediction methods such as Common Neighbors, Adamic-Adar, and Jaccard Coefficient to identify potential future links in a network. **(4 hours)**
2. Apply SimRank and PathSim to compute similarity scores for link prediction in graph-based data. **(2 hours)**
4. Compare heuristic and probabilistic link prediction models using evaluation metrics like precision and recall. **(2 hours)**
5. Use the Girvan–Newman algorithm to detect disjoint communities by iteratively removing high-betweenness edges. **(2 hours)**
6. Apply the Louvain or Leiden algorithm to uncover modular communities in large-scale networks. **(2 hours)**
7. Evaluate community detection results using modularity and normalized mutual information (NMI). **(2 hours)**
8. Simulate the Independent Cascade Model (ICM) to observe the spread of influence from a given set of seed nodes. **(2 hours)**
9. Simulate the Linear Threshold Model (LTM) to model influence propagation based on node activation thresholds. **(2 hours)**
10. Implement Greedy and CELF algorithms to efficiently select the top-k influential nodes for maximizing spread. **(4 hours)**
11. Benchmark different influence maximisation techniques, including simulation-based, sketch-based, and community-based methods. **(8 hours)**

DSE202: Combinatorial Optimization [3-0-1]

Course Code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice	
DSE202	**4**	**3**	**0**	**1**	**knowledge of Linear Algebra**

Course Objectives:

The course aims to equip students with the technique of linear and integer programs to solve optimization problems via LP-based solutions to shortest path problems, minimum spanning tree problems, max-flow problems and maximum matching problems.

Course Learning Outcomes:

On completion of this course, students will be able to:

1. model problems using linear and integer programs.

2. differentiate between the computational complexities of LP and IP.
3. understand polyhedral analysis and apply it to develop algorithms.
4. understand the concept of duality and use it to design exact and approximate algorithms.
5. understand and explain the mathematical theory forming the basis of many algorithms for combinatorial optimization (particularly graph theoretic).

Syllabus:

Unit-I **(15 hours)**
Introduction: Optimization problems, neighborhoods, local and global optima, convex sets and functions, simplex method, degeneracy; duality and dual simplex algorithm, computational considerations for the simplex and dual simplex algorithms-Dantzig-Wolfe algorithms.

Unit-II **(10 hours)**
Integer Linear Programming: Cutting plane algorithms, branch and bound technique and approximation algorithms for travelling salesman problem.

Unit-III **(15 hours)**
Graph Algorithms: Primal-Dual algorithm and its application to shortest path (Dijkstra's algorithm, Floyd-Warshall algorithms), max-flow problem (Ford and Fulkerson labeling algorithms), matching problems (bipartite matching algorithm, non-bipartite matching algorithms, bipartite weighted matching-hungarian method for the assignment problem, non-bipartite weighted matching problem), efficient spanning tree algorithms.

Unit-IV **(5 hours)**
Matroids: Independence Systems and Matroids, Duality, Matroid Intersection.

Readings:

1. Korte, Bernhard H., Jens Vygen, B. Korte, and J. Vygen. *Combinatorial optimization*. Springer, 2018.
2. Matoušek, Jiří, and Bernd Gärtner. *Understanding and using linear programming*. Springer, 2007.
3. Papadimitriou, Christos H., and Kenneth Steiglitz. *Combinatorial optimization: algorithms and complexity*. Dover Publication, 1998.
4. Bazaraa, Mokhtar S., John J. Jarvis, and Hanif D. Sherali. *Linear programming and network flows*. John Wiley & Sons, 2011.
5. Taha, Hamdy A. *Operations research: an introduction*. Pearson Education India, 2014.

List of Practicals:

1. WAP to implement Simplex Method **(4 hours)**
2. WAP to implement Dual Simplex Method **(6 hours)**

3. WAP to implement Primal-Dual algorithm for shortest path problem **(6 hours)**
4. WAP to implement Floyd-Warshall algorithm for shortest path problem **(6 hours)**
5. Implement algorithms for matching problems. **(8hours)**

DSE203: Cyber Security [3-0-1]

Course Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
DSE203	**4**	**3**	**0**	**1**		**Graduation**

Course Objectives:

This course will be responsible for laying the foundation for creating a comprehensive understanding and expertise in the field of cyber security. This paper will set the level field for all the students to be able to come at par and move together as they must go deeper into hard-core cyber security topics during the course duration.

Course Learning Outcomes:

On completion of this course, students will be able to

1. state the need and scope for cyber laws.
2. enumerate various network attacks, describe their sources, and mechanisms of prevention.
3. describe the genesis of SCADA policies and their implementation framework.
4. carry out malware analysis.

Syllabus:

Unit-I **(7 hours)**

Introduction: Cyberspace, Internet, Internet of things, Cyber Crimes, cybercriminals, Cyber Security, Cyber Security Threats, Cyber laws and legislation, Law Enforcement Roles and Responses.

Unit-II **(15 hours)**

Cyberspace Attacks: Network Threat Vectors, MITM, OWASP, ARP Spoofing, IP & MAC Spoofing, DNS Attacks, SYN Flooding attacks, UDP ping-pong and fraggle attacks, TCP port scanning and reflection attacks, DoS, DDOS. Network Penetration Testing Threat assessment, Penetration testing tools, Penetration testing, Vulnerability Analysis, Threat matrices, Firewall and IDS/IPS, Wireless networks, Wireless Fidelity (Wi-Fi), Wireless network security protocols, Nmap, Network fingerprinting, BackTrack, Metasploit.

Unit-III **(15 hours)**

Introduction to SCADA (supervisory control and data acquisition): Understanding SCADA security policies, SCADA Physical and Logical Security, understanding differences between physical and logical security, define perimeter controls and terms, define various security zones, understand communication cyber threats, understand firewall, architectures.

Unit-IV **(8 hours)**

Introduction Malware Analysis: Static Analysis, Code Review, Dynamic Analysis, Behavioral analysis of malicious executable, Sandbox Technologies, Reverse-engineering malware, Defeat anti-reverse engineering technique, automated analysis, intercepting network connections, Network flow analysis, Malicious Code Analysis, Network analysis.

Readings:

1. Peter W. Singer and Allan Friedman, *Cybersecurity and Cyberwar*, Oxford University Press, 2014.
2. Michael Sikorski, Andrew Honig, *Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software* 2012, No Starch Press, San Francisco.
3. Dejay, Murugan, *Cyber Forensics* Oxford university press India Edition, 2018.
4. R. Rajkumar, D. de. Niz and M. Klein, *Cyber Physical Systems*, Addision-Wesely, 2017
5. Jonathan Clough, *Principles of Cybercrime*, Cambridge University Press, 24-Sep-2015.

References:

1. CEH Official Certified Ethical Hacking Review Guide, Wiley India Edition, 2015.

List of practicals :

Disclaimer: These lab exercises are for educational purposes only. All activities should be performed within the boundaries of the law.

1. Identify your machine's IP configuration using *ipconfig (Windows)* or i*fconfig (Linux)*. Ping various local and remote hosts to verify connectivity, trace the route to a website using *tracert* or *traceroute,* perform DNS lookups using *nslookup*, and gather domain registration details using the *whois* tool. (2hours)
2. Use tools like John the Ripper and Hydra to crack passwords. The task will involve using wordlist-based and brute-force techniques to break into password-protected accounts or services. (4 hours)
3. SET tool is one of the tool available in Kali Linux. Explore the tool and list down all types of Social Engineering attacks available in the tool. Further, make a document mentioning the Name of the attack, how the attack can be launched, and how I can make my system safe from these attacks. (6 hours)
4. Use the Social Engineering Toolkit (SET) available in Kali Linux to clone a vulnerable website (e.g., http://www.itsecgames.com/). Compare the cloned website with the original, explore different social engineering attacks such as phishing or credential harvesting**. (4 hours)**
5. Use *theHarvester* tool to collect at least 25 email addresses from various publicly available sources on the internet. These addresses can later be used for phishing simulations or further social engineering attacks. (2 hours)
6. Utilize tools from the *iHunt intelligent framework* available on web to create a sock puppet account. Document the tools used, their functionality, and how this can be applied in a

simulated information-gathering exercise. **(2 hours)**

7. Using the email addresses collected in the lab 6, perform further information gathering. Use search engines and social media platforms to find publicly available information about the email owners. Document the results using tools from iHunt Framework for email investigation.**(2 hours)**

   Note: Don't actually contact the person of interest. Just gather information which available publicly. Don't break any law.

8. Conduct a vulnerability scan on target systems using *NMap*. Analyze the results to identify potential vulnerabilities such as open ports, weak configurations, and security loopholes. **(4 hours)**

9. Use the Metasploit framework to exploit vulnerabilities and gain access to a remote Linux machine. The exercise involves using Kali Linux with Metasploit to exploit a vulnerable system. **(4 hours)**

10. A lab exercise related to SCADA & Industrial Control Systems (ICS) Security **(4 hours)**

DSE204: Information Retrieval [3-0-1]

Course Code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice	
DSE204	**4**	**3**	**0**	**1**	**Basic understanding of Statistics and Probability is required.**

Course Objectives:

This course introduces the basics of Information Retrieval (IR), focusing on how information is organized, searched, and ranked. Students will learn about different search models, document processing techniques, and ways to measure search accuracy. The course also covers web search methods, including link analysis and web crawling.

Course Learning Outcomes:
Upon successful completion of this course, students will be able to

1. understand the fundamental concepts of Information Retrieval (IR), including information need, relevance, and early developments in IR systems.
2. apply different retrieval models, such as Boolean retrieval and ranked retrieval, to effectively search and organize information.
3. evaluate search performance using precision-recall, ranking measures, and other standard evaluation metrics.
4. process and represent documents using techniques like vector space modelling, feature selection, stemming, and similarity measures.
5. explore web search techniques, including link analysis methods like PageRank and HITS, as well as web crawling strategies.

Syllabus:

Unit-I **(15 hours)**
Introduction to Information Retrieval (IR), information, information need, and relevance, the IR system and its components, early developments in IR, user interfaces in IR, retrieval and IR models, Boolean retrieval, term vocabulary and postings list, ranked retrieval, inverted index, index construction, index compression.

Unit-II **(10 hours)**
Document processing, document representation techniques, vector space model, feature selection for IR, stop words, stemming, concept of document similarity, evaluation of information retrieval systems, notion of precision and recall, precision-recall curve.

Unit-III **(10 hours)**
Standard performance measures, MAP, reciprocal ranks, F-measure, NDCG, rank correlation, standard datasets for IR evaluation, web search and link analysis, web crawling techniques, link analysis methods, PageRank algorithm, HITS algorithm.

Unit-IV **(10 hours)**
Classification and Clustering: Notion of supervised and unsupervised algorithms, Naive Bayes, nearest neighbour and Rochio's algorithms for text classification, text clustering methods such as K-Means.

Readings:

1. Baeza-Yates, Ricardo, and Berthier Ribeiro-Neto. *Modern information retrieval*. Vol. 463, no. 1999. New York: ACM press, 1999.
2. Schütze, Hinrich, Christopher D. Manning, and Prabhakar Raghavan. *Introduction to information retrieval*. Vol. 39. Cambridge: Cambridge University Press, 2008.
3. Grossman, David A., and Ophir Frieder. *Information retrieval: Algorithms and heuristics*. Vol. 15. Springer Science & Business Media, 2004.

4. Buttcher, Stefan, Charles LA Clarke, and Gordon V. Cormack. *Information retrieval: Implementing and evaluating search engines*. Mit Press, 2016.
5. Croft, W. Bruce, Donald Metzler, and Trevor Strohman. *Search engines: Information retrieval in practice*. Vol. 520. Reading: Addison-Wesley, 2010.

List of Practicals

1. Design and implement a Boolean retrieval system that can process queries using AND, OR, and NOT operators on a given set of text documents. (**2 hours)**
2. Develop a text preprocessing pipeline that performs tokenization, case folding, stop-word removal, and stemming on a document corpus. (**2 hours)**
3. Write a program that constructs an inverted index for a collection of documents and supports retrieval of documents based on single-word queries, then implement index compression using variable-byte encoding and analyse the reduction in index size and its effect on retrieval performance. (**4 hours)**
4. Compute the Term Frequency-Inverse Document Frequency (TF-IDF) values for each term in a document collection and store them in a structured format. (**2 hours)**
5. Implement a ranked retrieval system using the vector space model and cosine similarity to score and rank documents based on a user query. (**2 hours)**
6. Use a publicly available IR dataset to implement and compare different retrieval models (e.g., Boolean, Vector Space), then calculate evaluation metrics including precision, recall, F1-score, MAP, and NDCG using provided relevance judgments, and finally, generate precision-recall curves for multiple queries to interpret the performance of each model. **(4 hours)**
7. Implement both the PageRank and HITS algorithms on a manually created web graph, simulate multiple iterations to observe rank convergence, and compute hub and authority scores for each node. (**4 hours)**
8. Train and evaluate a Naive Bayes classifier to categorize documents into predefined classes such as spam vs. ham or news categories. (**2 hours)**
9. Apply the K-Means clustering algorithm to a set of TF-IDF document vectors and visualize the resulting clusters using a 2D projection. (**4 hours)**
10. Develop a basic web crawler that extracts and stores titles and content from publicly accessible websites while respecting robots.txt. (**4 hours)**

DSE205: Digital Image Processing [3-0-1]

Course Code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical /Practice	
DSE 205	**4**	**3**	**0**	**1**	**Graduation**

Course Objectives:

The course will thoroughly introduce image enhancement in the spatial and frequency domain, followed by the image morphological operations such as dilation, erosion, hit-or-miss transformations, image segmentation and image compression.

Course Learning Outcomes:

Upon successful completion of this course, students will be able to

1. enhance the quality of an image using various transformations.
2. analyze the transform of an image in spatial domain to frequency domain.
3. apply required morphological operations to an image.
4. adequate to segment an image using various approaches.

Syllabus:

Unit-I **(7 Hours)**

Introduction: Applications of digital image processing, steps in digital image processing: image acquisition, image sampling and quantization, basic relationships between pixel.

Unit-II **(14 Hours)**

Image enhancement in the spatial domain and frequency domain: gray level transformations, histogram processing, local enhancement, image subtraction, image averaging, spatial filtering: smoothing and sharpening filters, Discrete Fourier transformation, filtering in the frequency domain: smoothening and sharpening filters, image restoration in spatial and frequency domains.

Unit-III **(12 Hours)**

Morphological image processing: erosion and dilation, opening and closing, hit-or-miss transformation, and some basic morphological algorithms.

Introduction to Image Compression: Image compression models, error free compression techniques, lossy compression techniques, JPEG, MPEG.

Unit-IV **(12 Hours)**

Image segmentation: Point, line and edge detection, gradient operator, edge linking and boundary detection, thresholding, region-based segmentation, representation schemes like chain codes, polygonal approximations, boundary segments, skeleton of a region, boundary descriptor, regional descriptor.

Readings:

1. Gonzalez, Rafael C. and Woods, Richard E., *Digital Image Processing*, 4th edition, Pearson Education, 2018.
2. Annadurai, S. and Shanmugalakshmi, R., *Fundamentals of Digital Image Processing*, 1st edition, Pearson, 2006.

3. Joshi, M. A., *Digital Image Processing: An Algorithmic Approach*, 2nd edition, PHI Learning, 2020.

References:

1. Jahne, Bernd, *Digital Image Processing*, 6th edition, Springer, 2005.
2. Chandra, B. and Majumder, D.D., *Digital Image Processing and Analysis*, 2nd edition, Prentice Hall India Learning Private Limited, 2011.

List of practicals:

You may choose a suitable programming language and all necessary/relevant inputs/ problems.

1. Implement a program for displaying grey-scale and RGB colour images, with those for accessing pixel locations, and investigate adding and subtracting a scalar value from an individual location. **(2 Hours)**

2. Implement a program to perform the image sampling and quantization technique and display the resulting images. **(4 Hours)**

3. Implement a program to apply histogram equalization to a colour image, and apply contrast stretching to the colour example image. Experiment with different parameter values to find an optimum for the visualization of this image. **(2 Hours)**

4. Implement a program to add different levels of salt and pepper and Gaussian noise to images, both in colour and grey-scale. Investigate the usefulness of all filtering for removing different levels of image noise. **(2 Hours)**

5. Write a program for all image restoration techniques. **(4 Hours)**

6. Write a program for the Fourier transform of an image. **(2 Hours)**

7. Write a program for image spatial and sharpening filtering. **(4 Hours)**

8. Implement a program to find the result of erosion, dilation, opening and closing with the below structuring element and choose any image. **(2 Hours)**

a)		**b)**
1 0 0		1 1 1
0 1 0		1 1 1
0 0 1		1 1 1

9. Write a program for all the compression techniques of an image. **(4 Hours)**

10. Write a program for all image segmentation techniques. **(4 Hours)**

DSE206: Advance Classification Methods [3-0-1]

Course Code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice	
DSE207	**4**	**3**	**0**	**1**	**Basic understanding of machine learning concepts, and familiarity with Python programming.**

Course Objectives

By the end of this course, students will be able to apply various classification techniques and evaluate model performance using appropriate metrics. They will also gain expertise in advanced topics such as ensemble learning, semi-supervised and self-supervised methods, zero-shot and few-shot learning, and improving model robustness against adversarial attacks.

Course Outcomes

On completion of this course, students will be able to:

1. explain the fundamental and advanced classification techniques.
2. apply various binary, multi-class, and multi-label classification methods.
3. analyze and compare classification models using appropriate evaluation metrics.
4. design and optimize classification models for real-world applications.

Syllabus

Unit-I **(16 Hours)**

Review of classification problems and algorithms, Evaluation metrics, Handling imbalanced data, Cross-validation, and Model selection. Ensemble learning for classification: Weak learners and Strong learners, Model creation, Model combination; Bagging and Boosting; Random Forest; Combination Strategies, Meta-learning; Diversity in ensemble; Ensemble pruning.

Unit-II **(12 hours)**

Multi-class classification problem; Aggregation-based approach: One-versus-all, One-versus-one, Directed Acyclic Graph approach, Tree-based approach, Error-correcting output codes; multi-class SVMs; Class Imbalance in Multi-class Settings.

Unit-III **(12 hours)**

Multi-label Classification, Problem Transformation approach: Binary Relevance, Classifier Chains, Label Powerset, Algorithm Adaptation Methods; Neural Network Approaches for Multi-label Classification, Evaluation metrics for multi-label classification.

Unit-IV **(5 hours)**

Semi-supervised and Self-supervised Classification Methods, Zero-shot and Few-shot Learning, Adversarial Attacks and Robustness in Classification,

Readings

1. Zhou, Zhi-Hua. *Machine learning*. Springer nature, 2021.
2. Mohri, Mehryar, A. Rostamizadeh, and A. Talwalkar. Foundations of machine learning. The MIT Press. 2012
3. Charte, Francisco, Antonio J. Rivera, and María J. Del Jesus. *Multilabel classification: problem analysis, metrics and techniques*. Springer International Publishing, 2016.
4. Ian Goodfellow, Bengio, Yoshua, and Aaron Courville. *Deep learning*. MIT press, 2017.
5. Research papers on emerging classification techniques.

List of Practicals :

1. Implement basic classification models such as Logistic Regression, Decision Trees and SVM on benchmark datasets. Evaluate the performance of these classifiers using metrics such as accuracy, precision, recall, F1-score, ROC-AUC, and confusion matrix. Perform k-fold cross-validation and interpret the results. **(2 hours)**
2. Use a dataset with imbalanced classes and then apply techniques like SMOTE, random oversampling/under sampling, and evaluate their impact on model performance using appropriate metrics like precision-recall and balanced accuracy. **(4 hours)**
3. Implement ensemble models, including Random Forest (bagging) and AdaBoost/Gradient Boosting (boosting), using scikit-learn or XGBoost. Compare performance, analyse feature importance, and visualise ensemble effects**. (4 hours)**
4. Create a diverse ensemble of traditional machine learning classifiers such as Logistic Regression, SVM and Decision Tree using stacking or voting. Analyze diversity among models and perform ensemble pruning based on accuracy or diversity criteria. **(4 hours)**
5. Use multi-class datasets and implement one-vs-all and one-vs-one strategies. Train binary classifiers accordingly and compare both strategies in terms of performance and scalability. **(2 hours)**
6. Implement multi-class classification using tree-based methods and error-correcting output codes (ECOC). Evaluate performance and interpret how each approach handles class boundaries. **(4 hours)**
7. Train and evaluate multi-class SVMs using one-vs-all or one-vs-one schemes. Use imbalanced versions of multi-class datasets to analyze the effect of class imbalance and apply weighted loss or resampling. (**2 hours)**
8. Implement multi-label classification approaches such as Binary Relevance, Classifier Chains, and Label Powerset using scikit-multilearn or similar libraries. Compare performance using hamming loss, precision, recall, and subset accuracy**. (4 hours)**
9. Build a neural network using PyTorch or similarly libraries for multi-label classification with appropriate activation function. Train the model on a multi-label dataset and evaluate using multi-label metrics. (**2 hours)**
10. Explore few-shot classification using pre-trained models and implement semi-supervised learning on a partially labelled dataset to improve performance with limited annotations. **(2 hours)**

DSE207: Computer Vision [3-0-1]

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical /Practice		
DSE 206	4	3	0	1		Graduation

Course Objectives:

The primary objective of this course is to introduce the fundamentals of image formation & representation and find the solution for the real-time challenges using deep learning.

Course Learning Outcomes:

Upon successful completion of this course, the student will be able to

1. recognize and describe both the theoretical and practical aspects of computing with images.
2. understand image transformations, segmentation, feature detection, motion, matching, recognition, extraction, and categorization from images.
3. understand the geometric relationships between 2D images and the 3D world.
4. analyze the performance of deep networks within an image.

Syllabus:

Unit I: (11 Hours)

Overview of Computer Vision and Image Processing: Introduction of computer vision, definition, applications, and importance; historical context and evolution; image acquisition, image formats and representations, image enhancement, image filtering and convolution, RGB, HSV, and other color spaces, color manipulation in images.

Unit II: (12 Hours)

Early and Mid-level Vision: translation, rotation, scaling; affine and perspective transformations, thresholding, region-based segmentation, segmentation by clustering, segmentation by fitting a model, segmentation and fitting using probabilistic methods, feature detection and matching: points and patches, edges, lines, edge detection, corner detection, histogram equalization, adaptive histogram equalization, template matching, scale-invariant feature transform (SIFT).

Unit III: (10 Hours)

High Level Vision: Geometric methods: model-based vision, smooth surfaces and their outlines, aspect graphs, range data; probabilistic and inferential methods: recognition by relations between templates, geometric templates from spatial relations, application, image-based rendering.

Unit IV: (12 Hours)

Dense Motion Estimation: Translational alignment, parametric motion, spline-based motion, optical flow, layered motion.

Deep learning-based Computer Vision Techniques: Deep learning for computer vision, basics of neural networks, convolutional neural networks (CNNs), architecture of CNNs, training and fine-tuning CNNs, LeNet, AlexNet, GooleNet, VGG-Net, ResNet, comparative analysis of different architecture.

Readings:

1. Gonzalez, Rafael C. and Woods, Richard E., *Digital Image Processing*, 4^{th} edition, Pearson Education, 2018.
2. Forsyth and Ponce, *Computer Vision: A Modern Approach*, 2^{nd} edition, Pearson, 2015.
3. Szeliski, Richard, *Computer Vision: Algorithms and Applications*, Springer-Verlag, 2010.
4. Prince, Simon J. D., *Computer Vision: Models, Learning, and Inference*, 1^{st} edition, Cambridge University Press, 2012.
5. Goodfellow, Ian, *Deep Learning (Adaptive Computation and Machine Learning series)*, The MIT Press, 2016

References:

1. Hartley, Richard and Zisserman, Andrew, *Multiple View Geometry in Computer Vision*, 2^{nd} edition, Cambridge University Press, 2004.

List of practicals:

You may choose a suitable programming language and the necessary/relevant inputs/ problems.

1. Implement a program for filtering and convolution techniques on the image. **(2 Hours)**

2. Implement a program to perform all colour space conversion and colour manipulation techniques in images. **(2 Hours)**

3. Write a program for all image translation, rotation, and scaling techniques. **(3 Hours)**

4. Write a program for all image segmentation techniques. **(4 Hours)**

5. Implement a program to perform all feature detection and matching techniques in images. **(3 Hours)**

6. Implement a program to apply histogram equalization to a colour image, and apply contrast stretching to the colour example image. Experiment with different parameter values to find an optimum for the visualization of this image. **(2 Hours)**

7. Write a program for the scale-invariant feature transform of an image. **(3 Hours)**

8. Implement a program for geometric methods of the image. **(3 Hours)**

9. Implement a program for dense motion estimation techniques on the image. **(2 Hours)**

10. Implement a program to apply different architectures of CNN to images to find the result of the image and investigate the usefulness of different architectures of CNN. **(6 Hours)**

List of GEs for Semester II

GE201: DATA MINING [3-0-1]

Course Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
GE201	**4**	**3**	**0**	**1**		**Graduation**

Course Objectives:

In this course, the objective is to introduce the KDD process. The course should enable students to translate real-world problems into predictive and descriptive tasks. The course also covers data cleaning and visualization and supervised and unsupervised mining techniques.

Course Learning Outcomes:

At the end of the course, students will be able to

1. distinguish between the process of knowledge discovery and Data Mining.
2. play with basic data exploration methods to develop understanding of given data
3. identify suitable pre-processing method for give problem.
4. describe different data mining tasks and algorithms.
5. use programming tools (e.g. Weka/Python/R etc) for solving data mining tasks.
6. follow formal notations and understand the mathematical concepts underlying data mining algorithms

Syllabus:

Unit-I **(9 hours)**

Overview: The process of knowledge discovery in databases, predictive and descriptive data mining techniques, and unsupervised learning techniques. Data preprocessing: Data cleaning, Data transformation, Data reduction, Discretization

Unit II **(12 hours)**

Classification: Supervised learning/mining tasks, Decision trees, Decision rules, Statistical (Bayesian) classification, Instance-based methods (nearest neighbor), Evaluation and Validation methods.

Unit III **(12 hours)**

Clustering: Basic issues in clustering, Partitioning methods (k-means, expectation maximization), Hierarchical methods for clustering, Density-based methods, Cluster Validation methods and metrics

Unit IV **(12 hours)**

Association Rule Mining: Frequent item set, Maximal and Closed item sets, Apriori property, Apriori algorithm.

Readings:

1. Han, Jiawei, Jian Pei, and Hanghang Tong. *Data Mining: Concepts and Techniques* 3rd edition. Morgan Kaufmann, 2022.
2. Zaki, Mohammed J., WM Jr, *Data Mining and Analysis: Fundamental Concepts and Algorithms,* Cambridge University Press, 2014.
3. Tan, Pang-Ning, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining*, Addison Wesley, 2006.
4. Charu, C. *Data Mining: The Textbook*, Springer, 2015

List of Practicals:

1. Data Preprocessing & Visualization (4 Hours)

- Load and explore real-world datasets using Python/Weka/R
- Perform data cleaning: handle missing values, detect/treat outliers
- Apply data transformation: normalization, encoding categorical variables
- Use data reduction techniques: PCA, sampling
- Apply discretization/binning on continuous features
- Visualize data using histograms, boxplots, scatter plots, and heatmaps

2. Supervised Learning – Classification (4 Hours)

- Implement a Decision Tree classifier and visualise the tree
- Apply Naive Bayes classifier and evaluate accuracy
- Implement k-Nearest Neighbor (k-NN) classifier with different values of *k*
- Evaluate classifiers using confusion matrix, precision, recall, and F1-score
- Perform k-fold cross-validation and plot ROC-AUC curves

3. Unsupervised Learning – Clustering (8 Hours)

- Apply k-Means clustering and determine optimal k using Elbow/Silhouette method
- Perform Agglomerative Hierarchical clustering and plot dendrogram
- Apply DBSCAN for density-based clustering and tune parameters (eps, minPts)
- Visualize and interpret clustering results

4. Association Rule Mining (4 Hours)

- Use Apriori algorithm to find frequent itemsets from transaction data
- Generate association rules using confidence and lift thresholds
- Analyze discovered rules for business decision insights

5. Mini-Project (10 Hours)

- Select a real-world dataset (e.g., healthcare, e-commerce, finance)
- Apply full KDD process: preprocessing, modelling, (classification/clustering/association), and evaluation

- Summarize findings with visualizations and model insights

GE202: Data Science using Python [3-0-1]

Course Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
GE202	**4**	**3**	**0**	**1**		**Graduation**

Course Description

This course provides an in-depth introduction to data science using Python, covering foundational libraries, statistics, machine learning, deep learning, and recommendation systems. The course blends theoretical concepts with hands-on projects using real-world datasets.

Course Learning Outcomes

By the end of the course, students will be able to

1. apply Python libraries for data manipulation and visualization.
2. apply statistical and probability concepts to data science problems.
3. build and evaluate machine learning models for regression and classification.
4. understand deep learning architectures like CNNs and Transformers.

Syllabus:

Unit-I **(10 hours)**

Python & Statistical Foundations for Data Science: *Python Foundations:* Introduction to Python for Data Science, Python libraries: NumPy, Pandas, Matplotlib and Seaborn, *Exploratory Data Analysis (EDA):* Summary Statistics & Data Inspection, Detecting Outliers (Z-Score, IQR), Correlation & Covariance; *Introduction to Statistics*: Descriptive vs. Inferential Statistics, Probability Distributions: Binomial, Poisson, Normal, Exponential, Bayes' Theorem & Conditional Probability, Confidence Intervals & Hypothesis Testing

Unit-II **(10 hours)**

Machine Learning – Unsupervised: Introduction to Clustering, K-Means Clustering: Elbow Method, Silhouette Score, Hierarchical Clustering: Agglomerative & Divisive, DBSCAN: Density-Based Clustering, Gaussian Mixture Models (GMMs), Evaluation metrics.

Unit-III **(11 hours)**

Machine Learning – Supervised Learning: Regression, Linear Regression & Polynomial Regression, Multiple Linear Regression, Bias and Variance, Regularisation Techniques, Classification, Logistic Regression, k-Nearest Neighbours (k-NN), Naïve Bayes Classifier, Decision Trees; Overfitting, Underfitting, Cross-Validation, Bootstrapping & Resampling.

Unit-IV: **(14 hours)**

Deep Learning & Recommendation Systems: Introduction to Neural Networks, Perceptron & Multi-Layer Perceptron (MLP), Activation Functions, Backpropagation & Optimization Techniques; Convolutional Neural Networks (CNNs), Convolution & Pooling Layers, Image Classification using CNNs, Transfer Learning with Pre-trained Models; *Transformers & NLP:* Introduction to Transformers, Self-Attention & Positional Encoding, Applications: Sentiment

Analysis, Named Entity Recognition; *Recommendation Systems:* Collaborative Filtering, Content-Based Filtering, Matrix Factorization: Singular Value Decomposition (SVD)

Readings:

1. McKinney, Wes. *Python for Data Analysis: Data Wrangling with Pandas, NumPy and iPython*, 2nd Ed., O'Reilly, 2017.
2. Tan, P., Steinbach, M., Karpatne, A., and Kumar,V. *Introduction to Data Mining*, 2nd Edition, Pearson Education, 2018.
3. Grolemund,G., Wickham, H., *R for Data Science*, 1st Ed., O'Reilly, 2017.
4. Goodfellow, Ian. *Deep Learning*, MIT Press, 2016.

List of Practicals:

A. ***Exploratory Data Analysis and Statistical Inference on Real-World Health Data:*** Download dataset from the World Health Organisation (WHO) Life Expectancy dataset or any open-source COVID-19 dataset and perform the following tasks: **(5 hours)**

1. Load and clean data using Pandas and use Matplotlib/Seaborn for visualisation of data.
2. Compute and save summary statistics and detect outliers using Z-score and IQR.
3. Analyse the correlation between life expectancy and socio-economic indicators.
4. Apply Bayes' Theorem to assess conditional probabilities (e.g., likelihood of high life expectancy given high GDP).
5. Perform hypothesis testing (e.g., test if life expectancy is significantly different between developed and developing countries).

B. ***Customer Segmentation using Unsupervised Learning:*** Apply clustering techniques to segment customers based on their purchasing behaviour. Download the E-commerce dataset: E-commerce customer data and perform the following tasks: **(5 hours)**

1. Normalize data using standard scaling.
2. Apply K-Means Clustering and find optimal k using the Elbow Method and Silhouette Score. Also, visualize clusters using PCA.
3. Perform Hierarchical Clustering and plot dendrograms.
4. Use DBSCAN to identify clusters, noise/outliers in the data.
5. Evaluate clustering performance using internal metrics (Silhouette, Davies-Bouldin).

C. ***Predicting House Prices using Regression Techniques:*** Build regression models to predict house prices. Download Boston Housing dataset or Ames Housing dataset. Perform the following tasks: **(8 hours)**

1. Perform EDA and feature selection.
2. Apply Linear Regression, Polynomial Regression, and Multiple Linear Regression.

3. Evaluate the bias-variance tradeoff.
4. Use Ridge and Lasso Regression to reduce overfitting.
5. Apply *k*-NN, Logistic Regression, Naïve Bayes, and Decision Trees for binary classification (e.g., whether house price is above average).
6. Evaluate models using Cross-Validation and metrics like RMSE, R^2, accuracy, precision, recall.

*D. **Deep Leaning for classification and regression:*** Train a CNN model to classify digits and to predict house process. **(8 hours)**

1. Design and train a 2D CNN model for classification of MNIST or ImageNet dataset.
2. Design and train a DL model for house price prediction using Boston Housing dataset or Ames Housing dataset.
3. Use Transfer Learning with CNN model to boost image classification for a small-sample sized dataset.

E. **Sentiment Analysis on Product Reviews using Transformers:** Analyze customer sentiments from reviews. Download Amazon or IMDB Reviews. Perform the following tasks: **(4 hours)**

1. Use a pre-trained Transformer model (like BERT) to perform **Sentiment Analysis** on reviews.

MASTER OF COMPUTER APPLICATIONS (MCA)

2-YEAR FULL TIME PROGRAMME

Post Graduate Curriculum Framework

under NEP 2020

(w.e.f 2025)

**DEPARTMENT OF COMPUTER
SCIENCE FACULTY OF
MATHEMATICAL SCIENCES
UNIVERSITY OF DELHI
DELHI
110007**

Master of Computer Application (MCA) Program Details

Affiliation

The proposed programme shall be governed by the Department of Computer Science, Faculty of Mathematical Sciences, University of Delhi, Delhi-110007.

Programme Structure and Objectives

The MCA is a four-semester program spanning two years. It has two structures - **MCA with Coursework, and MCA with Coursework and Internship.**

First three semesters comprise of course - work. In the fourth semester, students are required to do an industry project with at least one supervisor from the department.

The Programme objectives of MCA with Coursework are to

- **E**quip the students with the **foundations of** computer science.
- **Enable the students to follow the career path of their choice by choosing** courses from a wide list of specialised **c**ourses with progression**.**
- **Prepare the students to take up a career in the highly competitive IT industry with Applications of Computer Science.**

The Programme objectives of MCA with Coursework and Internship are to

- **E**quip the students with the **foundations of** computer science.
- **Enable the students to follow the career path of their choice by choosing** courses from a wide list of specialised **c**ourses with progression**.**
- Provide students **with hands-on experience on live projects**, fostering practical skills and enabling them to prepare technical reports.
- **Prepare the students to take up a career in the highly competitive IT industry with applications of Computer Science.**

Project: Each student shall carry out a project in the fourth semester. The Project will be carried out under the supervision of the teacher(s) of the department. When the project is carried out in an external organization (academic institution/ industry), a supervisor may also be appointed from the external organization. The project work will be evaluated jointly by the internal supervisor and an examiner to be appointed by the department in consultation with the internal supervisor.

The project evaluation shall be as follows:

- Mid-semester evaluation: 30% weight
- End-semester evaluation
    - Dissertation: 30%weight
    - Viva-voce: 40%weight

Semester I

Semester I					
	Number of core courses				**3**
Course Code	**Course Title**	**Credits in each core course**			
		Theory	Tutorial	Practical	**Total**
DSC101	Object Oriented Programming	3	0	1	**4**
DSC102	Data Structures	3	0	1	**4**
DSC103	Database Systems	3	0	1	**4**
SBC101	Software Tools and Techniques	0	0	2	**2**
	Total credits in core course				14
	Number of DSE/GE courses				2*
	DSE 1	3	0	1	**4**
	DSE2/ GE1	3	0	1	**4**
	Total credits in DSE/GE				8
	Total credits in Semester I				**22**

***Select two DSEs or one DSE and one GE**

List of DSEs for Semester I		
Course Code	**Course Title**	**L-T-P**
DSE101	Network Science	3-0-1
DSE102	Graph Theory	3-0-1
DSE103	Computer Organization and Architecture	3-0-1
DSE104	Java Programming	3-0-1
DSE105	Statistical Methods	3-0-1
DSE106	Web Technologies	3-0-1

Semester II					
	Number of core courses				**3**
Course Code	**Course Title**	**Credits in each core course**			
		Theory	Tutorial	Practical	**Total**
DSC201	Design and Analysis of Algorithms	3	0	1	**4**
DSC202	Operating Systems	3	0	1	**4**
DSC203	Artificial Intelligence and Machine Learning	3	0	1	**4**
SBC201	Scientific Writing and Computational Analysis Tools	0	0	2	**2**
	Total credits in core course				14
	Number of DSE/GE courses				2*
	DSE3	3	0	1	**4**
	DSE4 / GE2	3	0	1	**4**
	Total credits in DSE/GE				8
	Total credits in Semester II				**22**

***Select two DSEs or one DSE and one GE**

List of DSEs for Semester II		
Course Code	**Course Title**	**L-T-P**
DSE201	Social Networks Analysis	3-0-1
DSE202	Combinatorial Optimization	3-0-1
DSE203	Cyber Security	3-0-1
DSE204	Information Retrieval	3-0-1
DSE205	Digital Image Processing	3-0-1
DSE206	Data Mining	3-0-1

Semester III					
	Number of core courses				**2**
Course Code	**Course Title**	**Credits in each core course**			
		Theory	**Tutorial**	**Practical**	**Total**
DSC301	Data Communication and Computer Networks	3	0	1	**4**
DSC302	Information Security	3	0	1	**4**
SBC301	TBD	0	0	2	**2**
	Total credits in core course				10
	Number of DSE/GE courses				3*
	DSE 5	3	0	1	4
	DSE 6	3	0	1	4
	DSE 7 / GE 3	3	0	1	4
	Total credits in GE/DSE				**12**
	Total credits in Semester III				**22**

List of DSE for Semester III		
Course Code	**Course Title**	**L-T-P**
TBD		

Semester IV (Structure 1 Coursework)

Semester IV (Structure 1 Coursework)					
	Number of core courses				**2**
Course Code	**Course Title**	**Credits in each core course**			
		Theory	**Tutorial**	**Practical**	**Total**
DSC401	Software Engineering	3	0	1	**4**
DSC402	Deep Learning	3	0	1	**4**
SBC401	TBD	0	0	2	**2**
	Total credits in core course				10
	Number of DSE/GE courses				3*
	DSE 8	3	0	1	**4**
	DSE 9	3	0	1	**4**
	DSE 10 / GE 4	3	0	1	**4**
	Total credits in GE/DSE				12
	Total credits in Semester IV				**22**

List of DSE for Semester IV (Structure 1 Coursework)		
Course Code	**Course Title**	**L-T-P**
TBD		

Semester IV (Structure 2 Coursework+Internship)					
	Number of core courses				**2**
Course Code	**Course Title**	**Credits in each core course**			
		Theory	**Tutorial**	**Practical**	**Total**
DSC401	Internship	0	0	16	**16**
	Techniques of Report Writing	0	0	2	**2**
	Total credits in core course				18
	Number of DSE/GE courses				1*
	DSE 11 / GE 5	3	0	1	**4**
	Total credits in GE/DSE				4
	Total credits in Semester IV				**22**

List of DSE for Semester IV (Structure 2 Coursework+Internship)		
Course Code	**Course Title**	**L-T-P**
TBD		

The following outcomes must be achieved by the end of the fourth semester in addition to the outcomes mentioned for Structure I for **Structure 2 (Coursework+Internship).**

i) Completion of experimentation/ fieldwork or similar tasks.

ii) Submission of the project report.

SEMESTER - I

DSC101: OBJECT-ORIENTED PROGRAMMING [3-0-1]

Course Code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice	
DSC101	4	3	0	1	Graduation

Course Objectives:

The course aims to develop the student's problem-solving skills. The course also focuses on debugging skills. The student learns to develop modular, well-documented code.

Course Learning Outcomes:

On completing this course, the student will be able to:

1. select a suitable programming construct and in-built data structures for the given problem.
2. design, develop, document, and debug modular programs.
3. use recursion as a programming paradigm for problem-solving.
4. apply object-oriented paradigm for problem-solving.

Syllabus:

Unit-I **(12 hours)**

Notion of class, object, identifier, keyword, and literal; basic data types: int, float, string, Boolean; basic operators (arithmetic, relational, logical, assignment), standard libraries.

Unit-II **(12 hours)**

Modular program development, input and output statements, control statements: branching, looping, exit function, break, continue, and switch-break; arrays and pointers, testing and debugging a program.

Unit-III **(6 hours)**

Use of recursion as a programming paradigm for problem-solving.

Unit-IV **(15 hours)**

Object Oriented Programming Concepts: Use of classes, inheritance, and Polymorphism. Exception Handling and File Handling: Reading and writing text and structured files, errors and exceptions.

Readings:

1. R. G. Dromey, *How to Solve it by Computer*, Pearson, 2006.

2. Stanley B. Lippman, and Josée Lajoie, *C++ PRIMER*, Addison-Wesley, 2019.

3. Bjarne Stroustrup, *The C++ Programming Language (4th Edition)* Addison-Wesley, 2013

List of Practicals:

1. Write a program for **BMI Calculator** which accepts height and weight from the user and computes BMI value and category (underweight, normal, overweight, obese), where BMI is defined as the ratio of weight in kilograms by height in metres squared. **(2 hours)**

2. Write a program for **Electricity Bill Calculation** which accepts the number of units consumed and calculates the total bill amount using slab-based rates (e.g., ₹3/unit for first 100 units, ₹5/unit for next 100, ₹8/unit for above 200). It also prints the final bill with slab-wise charges. **(2 hours)**

3. Write a program that takes a paragraph from the user and calculates the number of words, sentences, and characters (excluding spaces), and identifies the longest word. **(2 hours)**

4. Write a program that accepts the marks of five subjects for a student, calculates the average, and assigns a grade based on a standard grading scale (A, if average>85, B if >70, C if >55, D if >40, F otherwise). **(2 hours)**

5. Write a menu-driven modular program for a **Contact Manager** with options to add, search, delete, and view contacts using loops and control statements. Choose appropriate data type to save details. **(2 hours)**

6. Write a program that accepts an array of integers from the user and a target value, then searches for the value using both linear and binary search functions and returns the index if found. **(2 hours)**

7. Write a recursive program for the following: **(4 hours)**

   - To compute the factorial of a given non-negative integer.
   - To find n^{th} number in the Fibonacci series
   - To solve the **Tower of Hanoi** problem for n disks, showing the steps to move disks from source to destination.
   - To add multiplies two number without using multiplication operator.

8. Write a program to implement a Student Record System using object-oriented programming concepts. Create a class called Student that includes attributes such as student ID, name, age, department, and marks. Define methods to input student details, store them in an appropriate data structure and display the details of a student based on their student ID. The program should allow the user to add multiple students and retrieve information of any specific student by searching with the ID. Use constructors for initializing student objects and demonstrate encapsulation by keeping attributes private and accessing them through getter methods. **(4 hours)**

9. Write a program to implement a Bank Account Management System using object-oriented programming principles. Begin by creating a base class called BankAccount

that includes attributes such as account number, account holder name, and balance. Define member functions to perform common operations like deposit(amount), withdraw(amount), and check_balance() for displaying the current balance.

Then, create two subclasses: SavingsAccount and CurrentAccount. In the SavingsAccount subclass, include additional rules such as maintaining a minimum balance (e.g., ₹1000), and deduct a penalty if the balance falls below the minimum. In the CurrentAccount subclass, implement features such as an overdraft limit (e.g., up to ₹5000), allowing withdrawals beyond the current balance up to the limit. Override appropriate methods to reflect the different rules in each account type.

Ensure that the program uses constructors for initialization and proper encapsulation of data members. The program should allow the user to create accounts, perform transactions (deposit/withdraw), and view account details through a menu-driven interface. **(4 hours)**

10. Write a program to model a Library Management System using object-oriented programming concepts. Define a class Book with attributes such as book_id, title, author, and availability status. Include methods for displaying book details, updating availability, and checking whether a book is currently issued or not.

Create a base class User that includes common attributes such as user_id, name, and borrowed_books. Then, define two subclasses Student and Faculty, which inherit from User. Implement method overriding to define different borrowing rules: for example, Student users can borrow up to 3 books for a maximum of 15 days, whereas Faculty users can borrow up to 5 books for a maximum of 30 days. Include a method borrow_book(book) and return_book(book) with logic to check borrowing limits and due date rules based on the user type.

Also, define a class BorrowedBook to store borrowing details like book_id, user_id, borrow_date, and due_date. Ensure the program maintains a list of all books and users and supports operations like: adding new books to the system, registering users, borrowing and returning books, and displaying user or book details.

The program should use constructors for initializing objects, apply encapsulation to restrict direct access to sensitive attributes, and demonstrate inheritance and method overriding effectively. You may also extend it with simple menus or command-line interaction for a smoother user experience. **(4 hours)**

11. Develop a file-based student database system where user input is stored in a text file and options are provided to add, view, and search records, with exception handling for input errors and missing data. **(4 hours)**

DSC102: DATA STRUCTURES [3-0-1]

Course Code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/Practice	
DSC102	4	3	0	1	Graduation

Course Objectives:

The course is designed to develop the student's ability to develop algorithms for creating and manipulating basic data structures. The students also learn to compare and contrast different data structures for the problem.

Course Learning Outcomes:

On completing this course, the student will be able to:

1. develop programs using fundamental data structures such as sets, lists, stacks, queues, trees, graphs, as well as advanced structures like balanced trees and heaps.
2. identify and apply the most appropriate data structure for solving specific computational problems efficiently.
3. analyze the time and space complexity of algorithms and data structures to make informed choices in program design.
4. optimize the use of data structures through suitable programming constructs to enhance performance across various scenarios.
5. implement and evaluate real-world applications using a combination of data structures, demonstrating problem-solving and critical thinking skills.

Syllabus:

Unit-I **(8 hours)**
Growth of functions, Recurrence relations: Asymptotic notations, solving recurrences using recursion trees.

Unit-II **(14 hours)**
Basic data Structures: Primitive Data Types, Abstract Data Types, Linear vs Non-Linear Data Structures, Arrays - Static and Dynamic, 2D Arrays, Linked Lists - Single, Doubly linked, Circular; Stacks and Queues using arrays and linked lists; operations, their analysis and applications.

Unit-III **(12 hours)**
Trees: Binary Tree, Tree Traversals, Binary Search Tree, Height Balanced Trees: AVL, Red-Black Trees, B and B+ Trees, Heaps, Priority Queues, operations, their analysis and applications.

Unit-IV **(11 hours)**
Sets: Sets, Multisets, Maps, Hash Tables, Dictionaries. Graphs: Representation of Graphs, Searching in Graphs – breadth first search and its applications, depth first search and its applications, Shortest paths, Spanning Trees.

Readings:

1. Goodrich, M., Tamassia, R. and Mount D, *Data Structures and Algorithms in C++/Java*, 2nd Edition, Wiley, 2016.
2. Elliot B. Koffman, Paul A.T. Wolfgang, *Objects, Abstraction, Data Structures and Design Using C++/Java*, 1st Edition, Wiley Global Education. 2005.
3. T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein, *Introduction to Algorithms*, 3rd Edition, Prentice-Hall of India Learning Pvt. Ltd. 2010.

List of Practicals:

1. Write a recursive function to solve a recurrence relation and manually build the recursion tree. Use code and time measurement to validate asymptotic complexity. **(4 hours)**
2. Create static and dynamic arrays, perform insertion, deletion, and resizing operations. Measure time complexity of each operation and explain memory overhead in dynamic arrays. **(6 hours)**
3. Implement all types of linked lists. Perform insertion and deletion at head, tail, and a given position. Demonstrate forward and backward traversal in doubly linked lists. **(8 hours)**
4. Build stack and queue data structures using arrays and linked lists. Perform push, pop, enqueue, and dequeue operations, and demonstrate applications like expression evaluation and palindrome checking. **(6 hours)**
5. Construct a binary tree from given input and perform preorder, inorder, and postorder traversals (both recursively and iteratively). **(6 hours)**
6. Create a BST and perform insert, delete, and search operations. Extend the implementation to AVL tree and demonstrate automatic balancing after operations. **(4 hours) (Optional)**
7. Create a hash table using open addressing and resolve collisions with linear and quadratic probing. Demonstrate performance with a load factor and compare efficiency. **(2 hours) (Optional)**
8. Implement graph using adjacency list/matrix. Perform BFS and DFS with applications like connected components and cycle detection. Use Dijkstra's algorithm for shortest paths and Kruskal's/Prim's for minimum spanning tree. **(4 hours) (Optional)**
9. Implement graph using adjacency list/matrix. Perform BFS and DFS with applications like connected components and cycle detection. Use Dijkstra's algorithm for shortest paths and Kruskal's/Prim's for minimum spanning tree. **(4 hours) (Optional)**

DSC103: DATABASE SYSTEMS [3-0-1]

Course Code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/Practice	
DSC103	4	3	0	1	Graduation

Course Objectives:

This course introduces the fundamentals of database systems, including data modeling, abstraction, and the DBMS architecture. It covers conceptual and relational database design using ER/EER models and integrity constraints. Students will learn to query and manipulate data using relational algebra, calculus, and SQL. Advanced topics like normalization, transaction management, and an introduction to NoSQL and XML databases are also explored.

Course Learning Outcomes:

On completion of this course, the student will be able to:

1. apply basic database concepts, including the structure and operation of the relational data model.
2. apply logical database design principles, including data normalization and E-R/EE-R diagrams to an application,
3. apply the integrity constraints to the application at hand.
4. answer database queries using Structured Query Language (SQL).
5. enumerate the concurrency control issues and identify their resolution strategies.
6. design and implement database projects.

Syllabus:

Unit-I **(6 Hours)**

Introduction to Database Systems: Data modelling for a database, abstraction and data integration, three-level DBMS architecture

Unit-II **(10 Hours)**

Database Design and Modelling: Entity-Relationship (ER) Model, Extended Entity-Relationship (EER) Model, Relational Model: Relations, Conversion of ER Diagrams to Relations, Integrity Constraints

Unit-III **(14 Hours)**

Querying and Manipulating Data: Relational Algebra, Relational Domain & Tuple Calculus, Structured Query Language (SQL): DDL, DML, Views, Embedded SQL

Unit-IV **(15 Hours)**

Advanced Database Concepts: Functional Dependencies, Determining Keys, Normalization, Lossless Join and Dependency-Preserving Decomposition, Transaction Management: ACID Properties, Concurrency Control, Transaction Recovery, Introduction to NoSQL and XML databases

Readings:

1. Silberschatz, H., Korth, S. and Sudarshan, S. *Database System Concepts*. 6th Edition, McGraw Hill, 2014.
2. Elmasri, R. and Navathe, S. B. *Fundamentals of Database Systems*. 7th Edition, Pearson, 2016.
3. Ramakrishnan, R. and Gehrke, J. *Database Management Systems*. 3rd Edition, McGraw Hill, 2014.
4. Lewis, P., Bernstein, A. and Kifer, M. *Databases and Transaction Processing – An Application-Oriented Approach*. Prentice Hall, 2003.

List of Practicals:

1. Define a schema for a real-world application with keys and constraints (e.g., Library/College System). Perform data manipulation using INSERT, UPDATE, DELETE, and SELECT statements in SQL. Use COUNT, SUM, AVG, MIN, MAX, GROUP BY, and HAVING. **(4 Hours)**

2. Model entities and relationships; map EER to tables. Perform multi-table queries using various types of JOIN and subqueries (correlated and nested). Apply and test NOT NULL, CHECK, DEFAULT, PRIMARY KEY, FOREIGN KEY, UNIQUE. **(4 Hours)**

3. Define simple and complex views; demonstrate access abstraction. Identify anomalies, functional dependencies, decompose tables, and implement a normalized schema in SQL. **(2 Hours)**

4. Interfacing SQL with Python programming language. Implement GUI-based environment using TkInter. **(4 Hours)**

5. Use START TRANSACTION, COMMIT, ROLLBACK, and simulate dirty read/lost update problems. Demonstrate locking behavior (e.g., using InnoDB), isolation levels, and test serializability. **(2 Hours)**

6. Choose a domain (e.g., Event Management, Hostel Management, E-commerce), build a full schema, write complex queries, use views, apply constraints, and simulate transactions. **(14 Hours)**

SBC101: SOFTWARE TOOLS AND TECHNIQUES [0-0-2]

Course Code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/Practice	
SBC101	**2**	**0**	**0**	**2**	

Course Objective:

The objective of this course is to develop proficiency in the use of software tools required for project development.

Course Learning Outcomes:

Upon successful completion of this course, students will

1. have knowledge of the fundamental shell tools and scripting techniques used in Linux or Unix-like environments.
2. understand the process of booting systems, using live USBs, and understanding their role in system recovery and troubleshooting.
3. apply Markdown for documentation, including its syntax and its applications in writing clear, readable documents.
4. Use Git for version control, including branching, merging, and resolving conflicts in

collaborative software development

Syllabus:

Unit I

Foundations of Shell and Command-line Environment: Shell Tools and Scripting, Editors (Vim), Command-line Environment, Debugging and Profiling, Common command-line flags/patterns, Booting + Live USBs, Metaprogramming: Working with Daemons, FUSE, Markdown

Unit II

Advanced Tools and Applications: Version Control (Git), Markdown language, Backup strategies: full, incremental, and differential, Vagrant and VMs, Docker and Containerization, Cloud Providers Overview, manage public and private clouds using OpenStack

Readings:

1. Newham C., *Learning the bash shell: Unix shell programming*. O'Reilly Media, Inc., 2005
2. Shotts W. *The Linux command line: a complete introduction*. No Starch Press. 2019.
3. https://git-scm.com/book/en/v2
4. https://www.techtarget.com/searchdatabackup/feature/Full-incremental-or-differential-How-to-choose-the-correct-backup-type
5. https://developer.hashicorp.com/vagrant/docs
6. https://docs.docker.com/get-started/
7. https://docs.openstack.org/2025.1/

List of Practicals

1. Shell Tools and Scripting: Write a script to display system information (uptime, disk usage, memory), use of variables, conditionals, and loops, Text Processing with Shell: Use grep, awk, sed, cut, sort, uniq, wc to process log files, count failed SSH login attempts from */var/log/auth.log*. **(4 hours)**

2. Vim Editor Mastery: Open a file, insert text, delete lines, use undo/redo, use search/replace, split windows, macros, and customise. vimrc. Command-line Environment: Bash Configuration and Aliases, Create and modify *.bashrc* or . *Zshrc* with aliases and functions, set environment variables, use history, jobs, fg, bg, kill, &amp, nohup. **(4 hours)**

3. Shell Script Debugging: Use bash -x and set -x to trace script execution, insert logging and echo for runtime info, Measure script runtime with time, perf, strace, and top, Common Command-line Flags/Patterns, explore tools like find, xargs, tee, tr, yes, watch, cut, paste, and pipeline, Batch rename files or extract columns from CSV using shell commands. **(4 hours)**

4. Creating and Using a Live USB: Create a bootable USB using dd, balenaEtcher, or Rufus, Boot from USB and explore the Linux live environment, Mount and access partitions, Exploring Boot Process, use dmesg, journalctl, systemctl, and lsblk to inspect boot logs and services. **(4 hours)**

5. Metaprogramming: Working with Daemons, FUSE, convert a script into a background

daemon process, manage it using systemctl or cron, Install FUSE, mount a simple FUSE filesystem (e.g., using hello.py or passthrough_fuse), Create a basic virtual FS using Python or C with fusepy. **(2 hours)**

6. Writing and Rendering Markdown: create a .md file with headings, lists, code blocks, links, and images. Convert Markdown to HTML/PDF using pandoc or preview with VSCode/Typora. Optional: Use grip or markdown-cli for local web preview. **(2 hours)**

7. Git Basics: Initialize a Git repository, Track files using git add, git commit, and git status, Branching and Merging, Create, switch, and merge branches, Solve a basic merge conflict. **(4 hours)**

8. Remote Repository and Collaboration, push to GitHub or GitLab, Clone, pull, and contribute using forks and pull requests. **(2 hours)**

9. Understand and implement Full and Incremental Backup strategies. Use tar or cp to create a full backup of the directory. **(2 hours)**

10. Schedule incremental backups using rsync or borg. Demonstrate how changes from the last full backup are saved using tools like rdiff-backup. **(2 hours)**

11. Vagrant and Virtual Machines (VMs): write a Vagrantfile to launch a Linux VM using VirtualBox and automate software installation using shell scripts or Ansible within the Vagrant VM. **(2 hours)**

12. Docker and Containerization: Install Docker on your system and verify the installation using the command *docker info*. Use *docker pull* to download official images of nginx and Ubuntu. Start a container and explore the nginx container by accessing it via the browser. Stop the container **(2 hours).**

13. In continuation of Lab exercise 11, write a Dockerfile to containerise a simple web app (e.g., Python Flask or Node.js) specifying dependencies and configurations. Build the Docker image and run the container. **(2 hours)**

14. Install *Docker Compose*. Write a simple docker-compose.yml file that defines two services: A web app (use Web App created in Lab exercise 12) and A database (e.g., MySQL or PostgreSQL). Demonstrate the Start and stop of both the web and database services. Further, show how Docker can be used to run multiple instances of the web app. **(4 hours)**

15. Explore the three leading cloud platforms: Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP). Further, make a document containing a comparison table that contrasts the main features and services provided by these three cloud service providers. Gather details on the following categories: Compute Services, Storage Services, Networking Services, Machine Learning and Analytics, and Security Services. **(4 hours)**

16. Deploying a Virtual Machine on AWS by creating an EC2 Instance on AWS. Select the right AMI (Amazon Machine Image), instance type, and configure security groups. Further, launch and connect to the EC2 instance created. **(2 hours)**

17. Deploying a Virtual Machine on Azure. Further, deploy a VM on Google Cloud Platform (GCP) using Compute Engine. Further explore the Google Cloud Console and basic VM settings (region, machine type, firewall rules). **(4 hours) (Optional)**

18. Networking Setup: Create Virtual Private Networks (VPCs) in AWS, Virtual Network in

Azure, and VPC in GCP. Configure subnets and firewall rules for VMs to communicate securely. **(2 hours) (Optional)**

19. Monitoring and Management: Introduce cloud-native monitoring tools for VM performance (e.g., CPU usage, network traffic). Set up basic alerts and logs for VM health and performance monitoring. **(2 hours)**

20. Cloud providers offer a wide variety of instance types optimised for different workloads (e.g., compute-optimised, memory-optimised, storage-optimised). Discuss AWS EC2 instance families such as T-series and M-series. Similarly, explore Azure's virtual machine types (e.g., A-series, D-series, E-Series. Review GCP machine types (e.g., N1-standard, C2, and M-series). Compare the trade-offs between choosing a general-purpose instance and specialised instances for specific workloads. **(6 hours)**

21. Build and Manage Public and Private Clouds using OpenStack: Install OpenStack using DevStack or Packstack on a VM or cloud host, launch instances, create networks, and assign floating IPs using Horizon (web UI) or CLI, write a script to automate instance creation using openstack CLI or heat templates. **(2 hours)**

22. Backup Containers and VM Snapshots: Demonstrate backing up Docker volumes and VM snapshots. Use Git + Docker + Vagrant + Cloud deployment for a CI/CD mini project. **(4 hours)**

List of DSEs for Semester I

DSE101: NETWORK SCIENCE [3-0-1]

Course Code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/Practice	
DSE101	4	3	0	1	Basic knowledge of probability theory

Course Objectives:

The course aims to acquaint the students with the graph theory concepts relevant for network science. The students learn dynamics of and on networks in the context of applications from disciplines like biology, sociology, and economics.

Course Learning Outcomes:

At the end of the course, the student will be

1. able to appreciate the ubiquity of the graph data model
2. able to understand the importance of graph-theoretic concepts in social network analysis
3. able to understand the structural features of a network
4. familiar with the theoretical graph generation models
5. identify community structures in networks
6. able to write programs to solve complex network problems

Syllabus:

Unit-I **(5 hours)**
Introduction to complex systems and networks, modelling of complex systems, review of graph theory.

Unit-II **(10 hours)**
Network properties: Local and global properties, clustering coefficient, All-pair-shortest path-based properties, centrality measures for directed and undirected networks, degree distribution, clustering coefficient.

Unit-III **(15 hours)**
Graph models: Random graph model, degree distribution, small world network model, power laws and scale-free networks, Barabasi-Albert (preferential attachment) network model, measuring preferential attachment

Unit IV **(15 hours)**
Community structure in networks: Communities and community detection in networks, Hierarchical algorithms for community detection, Modularity-based community detection algorithms, label propagation algorithm, multi-level graph.

Readings:

1. Pósfai, Márton, and Albert-László Barabási, Network Science, Cambridge University Press, 2016.
2. Newman, MEJ. Networks: An Introduction, Oxford University Press, 2010.
3. Easley David, and Jon Kleinberg, Networks, crowds, and markets: Reasoning about a highly connected world, Cambridge university press, 2010.
4. Meira Jr, W. A. G. N. E. R., and M. J. Zaki, Data mining and analysis, Fundamental Concepts and Algorithms, 2014.

List of Practicals:

1. Create and visualise a simple network and understand nodes, edges, and basic network structure. Use NetworkX (Python), Gephi. **(4 hours)**
2. Visualise Networks Using User Attributes. Use node color, size, or layout to reflect attributes like age or interest. **(2 hours)**
3. Implement Local and Global Network Properties. Also, measure average path length, diameter, and network density. **(2 hours)**
4. Calculate Node Centrality Measures and compare degree, closeness, betweenness, and eigenvector centrality. Use social graph e.g., Zachary's Karate Club and dolphin datasets. **(8 hours)**
5. Plot and analyze degree distributions in real-world and synthetic networks. **(2 hours)**
6. Implement Erdős–Rényi random graphs and assess their structural properties. **(2hours)**
7. Generate Small World Networks using Watts-Strogatz model and measure clustering and path length. **(2 hours)**
8. Implement Barabási–Albert Model and Generate preferential attachment networks and observe scale-free properties. (2 hours)
9. Implement and apply Hierarchical Community Detection algorithms to discover community structures. **(4 hours)**
10. Use the Louvain algorithm to find communities and optimize communities using modularity. **(2 hours)**

DSE102: Graph Theory [3-0-1]

Course Code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/Practice	
DSE102	4	3	0	1	Graduation

Course Objectives:

This course will thoroughly introduce the basic concepts of graphs theory, graph properties and formulations of typical graph problems. The student will learn to model which diverse applications in many areas of computing, social and natural sciences.

Course Learning Outcomes:

Upon successful completion of this course, the student will be able to

1. model problems using different types of basic graphs like trees, spanning tree, bipartite and planar graphs.
2. understand and identify special graphs like Euler graphs and Hamiltonian graphs.
3. have increased ability to understand various forms of connectedness in a graph.
4. appreciate different graph-coloring problems, matching problems and their solutions.

Syllabus:

Unit-I **(9 Hours)**

Fundamental Concepts: Definitions, examples of problems in graph theory, adjacency and incidence matrices, isomorphisms, paths, walks, cycles, components, cut-edges, cut-vertices, bipartite graphs, Eulerian graphs, Hamiltonian graphs, vertex degrees, reconstruction conjecture, extremal problems, degree sequences, directed graphs, de Bruijn cycles, orientations and tournaments, The Chinese postman problem.

Unit-II **(9 Hours)**

Trees: Trees and forests, characterizations of trees, spanning trees, radius and diameter, enumeration of trees, Cayley's formula, Prüfer code, counting spanning trees, deletion-contraction, the matrix tree theorem, graceful labeling, minimum spanning trees (Kruskal's algorithm), shortest paths (Dijkstra's algorithm).

Unit-III **(16 Hours)**

Matching and Covers: Matchings, maximal and maximum matchings, M-augmenting paths, Hall's theorem and consequences, Min-max theorems, maximum matchings and vertex covers, independent sets and edge covers, Connectivity, vertex cuts, Edge-connectivity.

Connectivity and Paths: Blocks, k-connected graphs, Menger's theorem, line graphs, dual graphs, network flow problems, flows and source/sink cuts, Ford-Fulkerson algorithm, Max-flow min-cut theorem.

Unit-IV **(11 Hours)**

Graph Coloring: Vertex colorings, bounds on chromatic numbers, Chromatic numbers of graphs constructed from smaller graphs, chromatic polynomials, properties of the chromatic polynomial, the deletion-contraction recurrence.

Planar Graphs: Planar graphs, Euler's formula, Kuratowski's theorem, five- and four-color theorems.

Readings:

1. West, Douglas B, *Introduction to Graph Theory*, 2nd Edition, Pearson, 2017.
2. Chartrand, Gary and Zhang, Ping, *Introduction to Graph Theory*, Tata McGraw Hill, 2017.
3. Gross, Jonathan L., Yellen, Jay and Anderson, Mark, *Graph Theory and Its Applications*, 3rd Edition, Taylor & Francis, 2024.

References:

1. Deo, Narsingh, *Graph Theory with Applications to Engineering and Computer Science*, Prentice Hall India Learning Private Limited, New edition, 1979.

List of practicals:

You may choose a suitable programming language and all necessary/relevant inputs/ problems.

1. Write a program to check whether the graph is: Bipartite, Eulerian, Hamiltonian or not. **(4 Hours)**

2. Write a program to find the minimum spanning trees of the graph. **(2 Hours)**

3. Write a program to find the shortest paths of the graph. **(2 Hours)**

4. Write a program to implement the Prüfer code of the graph. **(2 Hours)**

5. Write a program to implement the maximum and maximal matching of the graph. **(4 Hours)**

6. Write a program to implement the Ford-Fulkerson algorithm, Max-flow, and Min-cut theorem of the graph. **(6 Hours)**

7. Write a program to implement the Vertex colouring of the graph. **(2 Hours)**

8. Write a program to find the chromatic numbers of the graph. **(2 Hours)**

9. Write a program to check whether the graph is planar or not. **(2 Hours)**

10. Write a program to solve the Chinese postman problem. **(4 Hours)**

DSE103: COMPUTER ORGANIZATION AND ARCHITECTURE [3-0-1]

Course Code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice	
DSC103	4	3	0	1	Graduation

Course Objectives:

The course is designed to introduce basic building blocks of digital electronics, design and architecture of computer systems. This course aims to develop the skill to architect a digital computer system using a simulator.

Course Learning Outcomes:

On completing this course, the student will be able to:

1. describe the basic organization of computer hardware.

2. represent and manipulate data – number systems, conversion between different number systems, perform binary arithmetic.

3. design simple combinational and sequential logic circuits - flip-flops, counters, shift registers, adders, subtractor, multiplexer, demultiplexer, and Arithmetic/Logic unit.

4. design a CPU simple computer / microprocessor: instruction format, instruction set, addressing modes, bus structure, input/output architecture, memory unit, Arithmetic/Logic and control unit, data, instruction and address flow.

Syllabus:

Unit-I **(10 hours)**

Basic Building Blocks: Boolean logic and Boolean algebra, tri-state logic; flip-flops, counters, shift registers, adders, subtractors, encoders, decoders, multiplexers, demultiplexers.

Unit-II **(20 hours)**

Processor Design: CPU organisation, register organisation, stack organisation, microprogrammed control unit, RISC architecture; microprocessor architecture, modern computing architectures. Bus and memory transfers, arithmetic, logic shift micro-operations; basic computer organization: common bus system, instruction formats, instruction cycle, interrupt cycle, input/output configuration.

Unit-III **(6 hours)**

Memory Unit: Primary memory, secondary memory, associative memory, sequential access, direct access storage devices.

Unit-IV **(9 hours)**

Input-Output Architecture: Input/output devices; data transfer schemes - programmed I/O and DMA transfer; data transfer schemes for microprocessors.

Readings:

1. M. Morris Mano, *Computer System Architecture*, Revised 3rd Edition, Pearson, 2018.
2. W. Stallings, *Computer Organization and Architecture: Designing for Performance*, 9th Edition, Pearson Education, 2012.
3. A.S. Tanenbaum, *Structured Computer Organization*, 6th Edition, Prentice-Hall of India, 2012.
2. J.P. Hayes, *Computer System Architecture & Organization*, 3rd Edition, McGraw-Hill Education, 2017.

List of Practicals:

1. Design the circuits for Half Adder, Half Subtractor, Full Adder, and Full Subtractor using basic logic gates. **(4 hours)**

2. Design and simulate a multiplexer and demultiplexer (16:1 MUX, 1:16 DEMUX). **(2 hours)**

3. Implement encoders and decoders (e.g., 3-to-8 decoder, 8-to-3 encoder) using logic gates. **(2 hours)**

4. Design asynchronous and synchronous counters using flip-flops. **(2 hours)**

5. Design a common Bus system for four registers of size 4-bits using multiplexer and tri-state buffers. **(4 hours)**

6. Design and simulate circuits for the following Arithmetic Microoperations: **(6 hours)**

- Binary Adder
- Binary Adder-Subtractor
- Binary Incrementer
- Binary decrementer
- 4-bit Arithmetic Circuit for all the Arithmetic Microoperations

7. Design a logic circuit that performs basic logical operations like AND, OR, NOT, and XOR. **(2 hours)**

8. Design the circuits for arithmetic and logical shift microoperations. **(4 hours)**

9.Design and simulate a circuit for complete general register set architecture with common ALU. **(4 hours)**

DSE104: JAVA PROGRAMMING [3-0-1]

Course Code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/Practice	
DSE104	4	3	0	1	Graduation

Course Objectives:

The objective of this course is to develop skills in object-oriented concepts, writing programs using exception handling techniques, multithreading and user interface design of Java programming.

Course Learning Outcomes:

Upon successful completion of this course, the student will be able to

1. understand the object-oriented concepts, like Classes, Objects, Inheritance, Polymorphism and exceptions.
2. design, implement, document, test, and debug a Java application consisting of multiple classes and multithreading with input/output through files.
3. create Java applications with a graphical user interface (GUI).

Syllabus:

Unit-I **(7 Hours)**
Introductory Concepts: program, identifiers, variables, constants, primitive data types, expressions, control statements, structured data types, arrays, functions.

Unit-II **(10 Hours)**

Object Oriented Concepts: Abstraction, encapsulation, objects, classes, methods, constructors, inheritance, polymorphism, static and dynamic binding, overloading, Abstract classes, Interfaces and Packages.

Unit-III **(9 Hours)**

Exception handling: Throw and Exception, Throw, try and catch Blocks, Multiple Catch Blocks, Finally Clause, Throwable Class, Types of Exceptions, java.lang Exceptions, Built-In Exceptions.

Unit-IV **(19 Hours)**

File Handling: Byte Stream, Character Stream, File I/O Basics, File Operations, Serialization.

GUI Design: GUI based I/O, Input and Message Dialog boxes, Swing components, Displaying text and images in window.

Readings:

1. James Gosling, Bill Joy, Guy L. Steele Jr, Gilad Bracha, Alex Buckley, *The Java Language Specification*, Java SE 7 Edition, Addison-Wesley, 2013.

2. Cay S. Horstmann, *Core Java - Vol. I – Fundamentals*, 10th Edition, Pearson, 2017.

3. Deitel & Deitel, *Java-How to Program*, 9th Edition, Pearson Education, 2012.

References:

1. Richard Johnson, *An Introduction to Java Programming and Object-Oriented Application Development*, Thomson Learning, 2006.

2. Herbert Schildt, Java: *The Complete Reference*, 10th Edition, McGraw-Hill Education, 2018.

List of practical:

You may choose all necessary/relevant inputs/ problems.

1. Write a java program to design a class account using inheritance and static that shows all the functions (withdrawal and deposit) of the bank. **(2 Hours)**

2. Write a java program for method overloading and constructor overloading. **(2 Hours)**

3.Write a java program for dynamic method dispatch and final as class, variables, and methods. **(4 Hours)**

4. Write a java program to represent abstract class with example. **(2 Hours)**

5. Write a java program to implement inner classes and interface using extends keyword. **(2 Hours)**

6. Write a java program to create a user-defined package. **(2 Hours)**

7. Write a java program to implement exception handling techniques. **(4 Hours)**

8. Write a java program for the producer and consumer problem using threads. **(2 Hours)**

9. Write a java program to display the file class properties and load phone numbers with names from a text file. **(2 Hours)**

10. Write a java program to compute the factorial value using an Applet. **(2 Hours)**

11. Write a java program for handling Mouse events and Key events. **(4 Hours)**

12. Write a java program that works as a simple calculator. Use a Grid Layout to arrange Buttons for digits and for the +, -, *, % operations. Add a text field to display the result. **(2 Hours)**

DSE105: Statistical Methods [3-0-1]

Course Code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/Practice	
DSE105	4	3	0	1	Graduation

Course Objectives:

This course aims to equip students with the skills necessary to apply statistical methods for various applications.

Course Learning Outcomes:

On completing this course, the student will be able to:

1. apply descriptive statistical techniques to summarize and interpret data
2. apply inferential statistical methods, including hypothesis testing and confidence interval estimation.
3. perform and interpret simple and multiple linear regression analysis
4. apply principles of experimental design in the context of a problem

Syllabus:

Unit-I **(10 hours)**
Introduction: Descriptive statistics: measures of central tendency and variability, representation of data: stem and leaf diagram, histogram, boxplot, and ogive; bar diagram and its variations, Pie charts.

Unit-II **(10 hours)**
Probability distributions: discrete and continuous, joint and conditional probability; theory of attributes: coefficient of association and coefficient of colligation.

Unit-III **(15 hours)**
Statistical Inference: Parameter and statistic; sampling distributions, confidence intervals and margin of error, hypothesis testing; non-parametric inference: non-parametric tests: Mann-Whitney U test, Kruskal-Wallis test, Spearman's rank correlation coefficient.

Unit-IV **(10 hours)**
Regression and Classification: Correlation: measure and significance, simple linear regression,

multiple linear regression, one-way classification, analysis of variance, two-way classification, analysis of covariance, curvilinear regression, factorial experiments.

Readings:

1. Robert S. Witte and John S. Witte, *Statistics,* John Wiley & Sons Inc; 11th edition, 2021
2. Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani, *An Introduction to Statistical Learning*, Springer, 2023.
3. G. W. Snedecor, W. G. Cochran, *Statistical Methods*, Iowa State University Press, 1973
4. John A. Rice, *Mathematical Statistics and Data Analysis*, Cengage, 2013

List of Practicals :

Write the Python program for the following-

Refer research paper: Kumar, M. S., & Sahu, P. P. (2013). Employment Growth, Education and Skills in India: Emerging Perspectives. *Indian Journal of Labour Economics*, *56*(1), 95-122.

1. Select any data table and a variable from the selected data table. Compute central tendencies: arithmetic mean, geometric mean, median and standard deviation for the selected variable. **(4 hours)**
2. Refer table 9 for data of states for the year 1983 of primary sector for the persons who are educated up to middle level. Plot a box plot. Do same exercise for the year 1993-94 and comment on the shape of both box plots **(4 hours)**
3. Refer total population of various years in table 2: Plot an Ogive and Pie-chart (4 hours)
4. Refer table 4: is there any significant impact of gender on the education level? Test it on 95% confidence using Chi-square test **(4 hours)**
5. Refer table 5: From year 1983 to 2009-10, check the impact of time on non-literacy at 5% level of significance using Mann-Whitney U test **(4 hours)**
6. Refer table 5: From year 1983 to 2009-10, check the impact of time on non-literacy at 5% level of significance using Kruskal-Wallis test **(4 hours)**
7. Refer the data of 1994 and 2000 of table 2. Using Spearman's rank correlation, check whether data of 1994 and 2000 is strongly correlated? **(2 hours)**
8. Select a table and variables of your choice. Test the impact of treatment factor at 5% level of significance using one-way analysis of variance **(4 hours)**

DSE106: WEB TECHNOLOGIES [3-0-1]

Course Code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice	
DSE106	4	3	0	1	

Course Objectives:

This course provides a comprehensive understanding of web technologies and their applications, covering networking concepts, front-end development with modern frameworks, client-side programming with JavaScript, and server-side programming using Django. Additionally, Students will learn how to develop, test, and deploy dynamic web applications efficiently.

Course Learning Outcomes:

On completion of this course, students will be able to

1. understand web technologies concepts and networking protocols significance
2. build responsive web interfaces using HTML5, CSS, and frameworks like Bootstrap and React.js with a mobile-first approach.
3. apply various client-side web technologies for front-end development.
4. create server-side applications with Django, handling routing, authentication, databases, and REST APIs.
5. apply CI/CD practices to automate the development, testing, and deployment of web applications.

Syllabus:

Unit-I **(11 hours)**

Introduction to Networking and Web Technologies: Internet and its Evolution, World Wide Web, Web 2.0, Web 3.0, Network Communication Protocols: HTTP/HTTPS, SMTP, IMAP, POP, FTP, Client-Server Architecture, Web Applications Architecture, Application and Web Servers, Web Clients, Cloud Technologies: AWS, Azure, Google Cloud, API-First Architecture, RESTful APIs, WebSockets, Real-time Communication, Microservices Architecture, DevOps Concepts: Continuous Integration/Continuous Deployment (CI/CD) Pipelines

Unit-II **(10 hours)**

Front-end Development: Introduction to HTML5, HTML elements, HTML tags, lists, tables, frames, forms, Basics of XHTML, CSS Style Sheets, Responsive Design and Mobile-First Development, CSS Frameworks: Bootstrap, Materialize, Modern JavaScript Frameworks: Introduction to React.js, Angular, or Vue.js, Component-Based Architecture, CSS Preprocessors: SASS, LESS,, Front-end Testing: Jest, Mocha, Web Performance Optimization: Lazy Loading, Code Splitting

Unit-III **(12 hours)**

Client-Side Programming: Modern JavaScript (ES6+): Arrow Functions, Async/Await, Destructuring, Template Literals, Variables and Data Types, Literals, Functions, Objects, Arrays, Built-in Objects and Event Handling, Modifying Element Style, Document Trees, Advanced JavaScript Features: Closures, Promises, Async/Await, TypeScript Introduction, Single-Page Application (SPA) Development with React.js, State Management with Redux (for React) or Vuex (for Vue.js), Integrating APIs in React or Vue for dynamic content.

Unit-IV **(12 hours)**

Server-Side Programming: Introduction to Web Frameworks: Overview of Django, Setting up a Django Project, Routing and Views in Django, Templates in Django, Working with Forms in Django, Handling User Authentication and Authorization in Django, Authentication and Authorization: JWT (JSON Web Tokens), OAuth2.0, Introduction to Databases: Using SQLite, PostgreSQL, or MySQL with Django, CRUD Operations with Django ORM, REST APIs with Django Rest Framework, Session and Cookie Management in Django, Deployment of Django Applications

Resources:

1. Resources for developers by developers (Mozilla Firefox) https://developer.mozilla.org/en-US/
2. The OdinProject, https://www.theodinproject.com/
3. React: The library for web and native user interfaces, https://react.dev/
4. Django: the web framework for perfectionists with deadlines, https://www.djangoproject.com/
5. Scrimba: Helping developers learn faster by merging video and coding into one, https://scrimba.com/

List of Practicals:

The student will choose a project (website to be developed) of their choice and will do the following.

1. Initialise the project repository in GitHub and define the project scope in the README. **(2 hours)**
2. Create wireframes using Figma or some other tool to design a basic UI structure. **(4 hours)**
3. Develop 2-3 static web pages using HTML and CSS. **(2 hours)**
4. Implement basic client-side functionality such as whether the email entered format is correct or not, strength of the password etc **(4 hours)**
5. Set up a React.js project and create simple components. **(4 hours)**
6. Develop a few reusable components and handle state management depending on the project requirement **(4 hours)**
7. Set up a backend framework using Django **(4 hours)**
8. Create user login and registration features using Django **(4 hours)**

9. Set up and configure a relational database. Further, connect the React.js front-end with backend APIs. Integrate external APIs into the application, if any. **(4 hours)**
10. Deploy the project to a cloud platform of your choice. (additional)

SEMESTER – II

DSC201: DESIGN AND ANALYSIS OF ALGORITHMS [3-0-1]

Course Code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice	
DSC201	4	3	0	1	Graduation

Course Objectives:

The course introduces the techniques for algorithm design and analysis of the asymptotic performance of these algorithms. This course aims to achieve competence in designing efficient algorithms using different data structures for real-world problems.

Course Learning Outcomes:

On completing this course, the student will be able to:

1. describe various algorithm design techniques, including iteration, divide and conquer, dynamic programming, and greedy approach algorithms.
2. analyze the strengths and weaknesses of each technique.
3. identify and apply technique(s) suitable for simple applications.
4. demonstrate the correctness of algorithms and analyze their time complexity theoretically and practically.
5. model simple problems as graphs and solve them using Graph Algorithms.

Syllabus:

Unit-I **(4 hours)**
Computational Complexity: Review of Growth of Functions, asymptotic notation, Master's Theorem.

Unit-II **(20 hours)**
Searching and Sorting Techniques - Linear search, Binary search, insertion sort – time complexity. Binary Search, Merge sort and Quick sort – time complexity. Lower bounding techniques: Decision Trees.Linear Sorting: Count Sort, Radix Sort, Bucket Sort. Randomized algorithms: Introduction to random numbers, randomized Qsort, randomly built BST

Unit-III **(12 hours)**
Optimization Techniques: Greedy Algorithms: Interval Scheduling, Minimum Spanning Trees – Prim's algorithm, Kruskal Algorithm, Shortest Path Problem – Djikstra's algorithm. Dynamic Programming: Weighted Interval Scheduling, Matrix chain multiplication, Knapsack problem, Longest common subsequence.

Unit-IV **(9 hours)**
String Processing: Brute-force method, KMP algorithm. Introduction to class NP, NP-Hard and NP-Complete.

Readings:

1. Kleinberg, J. and Tardos,E. *Algorithm Design*. 1st Edition, Pearson Education India, 2013.
2. Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, *Introduction to Algorithms*, 4th Edition, The MIT Press, 2022.
3. Sara Baase, Allen Van Gelder, *Computer Algorithm – Introduction to Design and Analysis*, 3rd edition, 2009, Pearson Education.

Practicals

General Instructions:
1. All programs to be developed/uploaded immediately on GitHub.
2. Results obtained in a practical may be required in subsequent practical's. So, save them and keep them handy.
3. A program must be completely automated with no manual intervention - from taking the input values like n, m to generating the records of the results. Generating data sets can be internal to the program. It will be better to save it in a file also, in case it is required to be reused.
4. Time should not include the time for reading the input and writing the output.
5. If GenAI tools are used to create a part of the code, prior permission must be sought in writing, and it must be recorded as documentation at the top of the program.
6. Libraries for basic data structures like Stacks, Queues, Binary Search trees, Hashmaps can be used. For any other specialised part, prior permission must be sought.

List of Practicals

1. Implement Insert Sort and run it on a synthetic data of (Name : string type, age : float type) for n = 10, 20, …100. For each n, randomly generate at least 10 data sets. Report the average number of comparisons for each n. Plot a graph and obtain a best-fit curve. **(6 hours)**
    a. write a function to sort on age
    b. write a function to sort on names
    c. sort on age and then on names so that the ordering on age is maintained. i.e. For (Reeta, 18.3), (Reeta 17.8), (Geet, 18.3), the output should be (Geet, 18.3), (Reeta 17.8), (Reeta, 18.3). I.e. If there are two people with the name Reeta their records should appear so that they are sorted on their ages.

2. Program based on real life Application of Sorting: for example weather data or sort the edges of a graph on its weights - used in connectivity planning. **(2 hours)**
3. Repeat 1 and 2 with Merge Sort and Compare with the results of Insertion Sort. **(4 hours)**
4. Repeat 1 and 2 with Quick Sort and Compare with the results of Insertion Sort and Merge Sort. **(4 hours)**
5. Repeat 1 and 2 with Randomized Quick Sort and Compare with the results of Quick Sort. **(2 hours)**

6. Implement radix Sort and run it on a synthetic data of (Name : string consisting of 10 characters, age in years: integers in the range 10 to 20) for n = 10, 20, …100. For each n, randomly generate at least 10 data sets. Report the average number of comparisons for each n. Plot a graph and obtain a best-fit curve. **(4 hours)**
    a. write a function to sort on age
    b. write a function to sort on names
    c. sort on age and then on names so that the ordering on age is maintained.
    d. Compare with the results of Insertion Sort, Merge Sort, Quicksort and Randomized Quick Sort.

7. Programs on Graphs, as required. **(2 hours)**

8. Problem Solving based on Sorting, Searching, Graph problems, Greedy and Dynamic Programming. **(6 hours)**

DSC202: OPERATING SYSTEMS [3-0-1]

Course Code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/Practice	
DSC202	4	3	0	1	Graduation

Course Objectives:

This course aims to provide a functional perspective of the operating systems and develop skills to experiment with different components, tasks and services, including job scheduling, memory management, device management, process management and file management.

Course Learning Outcomes:

On completing this course, the student will be able to:

1. describe the basic functions of an Operating System.
2. distinguish between different types of operating systems to use each of them most efficiently in the respective application areas.
3. describe different techniques for managing computer resources like CPU, memory, file and devices.
4. implement algorithms for managing computer resources.
5. experiment with different components, tasks and services of the operating system.

Syllabus:

Unit-I **(5 hours)**
Introduction: Defining an Operating System, Design Goals, Evolutionary history of operating systems; Concept of User, job and Resources; Batch processing, Multi-programming, Time sharing; Structure and Functions of Operating System.

Unit-II **(18 hours)**

Process Management and Synchronization: Process states, State Transitions, Process Control Structure, Context Switching, Process Scheduling, Threads. Process Interaction, Shared Data and Critical Section, Mutual Exclusion, Busy form of waiting, Lock and unlock primitives, Synchronization, Classical Problems of Synchronization, Semaphores, Monitors, Conditional Critical Regions, System Deadlock, Wait for Graph, Deadlock Handling Techniques: Prevention, Avoidance, Detection and Recovery.

Unit III **(15 hours)**

Memory Management: Address Binding, Dynamic Loading and Linking Concepts, Logical and Physical Addresses, Contiguous Allocation, Fragmentation, Paging, Segmentation, Combined Systems, Virtual Memory, Demand Paging, Page fault, Page replacement algorithms, Global Vs Local Allocation, Thrashing, Working Set Model, Pre-Paging.

Unit IV **(7 hours)**

File and Secondary Storage Management: File Attributes, File Types, File Access Methods, Directory Structure, File System Organization and Mounting, Allocation Methods, Free Space management; Disk Structure, Logical and Physical View, Disk Head Scheduling, Formatting, Swap Management.

Readings:

1. Silberschatz, Galvin,and Gagne, *Operating Systems concepts*, Wiley, 2009.
2. Gary Nutt, Nabendu Chaki, Sarmistha Neogy, *Operating Systems: A Modern Approach* (3rd ed.), Addison Wesley, 2009.
3. Tanenbaum, Andrew S., and Herbert Bos. *Modern operating systems*. Pearson Education, 2015.
4. Stallings, William. *Operating systems: internals and design principles*. Prentice Hall Press, 2011.
2. D.M. Dhamdhere, *Operating Systems: A Concept Based Approach* (2nd ed.), Tata McGrawHill, 2007.

List of Practicals:

1. Write a C program that demonstrates process creation using fork (), executes a new program using exec(), and waits for child process termination using wait() system call. **(6 hours)**

2. Implement multiple threading using the thread library. Ensure graceful thread termination using join() to prevent abnormal exit. **(4 hours)**

3. Design a multithreaded program using pthreads and semaphores to simulate and demonstrate both mutual exclusion and deadlock scenarios. **(4 hours)**

4. Implement a race condition and then resolves it using: **(8 hours)**
 -pthread_mutex_lock() and pthread_mutex_trylock()
 -POSIX semaphores
 - Condition variables
 - Reader-writer locks (pthread_rwlock).

5. Develop a solution to the classic producer-consumer problem using a bounded buffer. Use mutexes for mutual exclusion and semaphores to synchronise producer and consumer threads.

(4 hours)

6. Write a C program that demonstrates interprocess communication between a parent and child process using unnamed pipes (pipe () system call). **(2 hours)**

7. Write a program that demonstrates file reading and writing operations. **(2 hours)**

DSC203: ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING [3-0-1]

Course Code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice	
DSC203	4	3	0	1	Basics of probability and statistics

Course Objectives:

Beginning with a comprehensive overview of the AI techniques, the course introduces the supervised and unsupervised ML techniques, along with their applications in solving real-world problems. The course also covers evaluation and validation methods for ML models.

Course Learning Outcomes:

On completion of this course, the student will be able to:

1. Understand foundational AI concepts, intelligent agents, logic systems, and classical search techniques.
2. Explore key machine learning models including regression, classification, and clustering algorithms.
3. Analyze model performance through bias-variance tradeoff, feature selection, and dimensionality reduction.
4. Learn artificial neural network architectures, training procedures, and optimization techniques.
5. Gain insights into advanced neural models including RNNs, SOMs, and Generative AI techniques.

Syllabus:

Unit-I **(15 hours)**
Introduction, Evolution, Application of AI, AI Problems, Problem Formulation, Intelligent Agents, Reasoning and Logic, Propositional Logic, First-order Logic, Inference in First-order Logic, Forward and Backward Chaining, Expert System, Solving Problems by Searching, Search - Uninformed Search Techniques, Heuristic Search Techniques, Advanced Search Techniques.

Unit-II **(10 hours)**
Machine Learning: Introduction, Supervised vs. Unsupervised learning, Regression: Linear regression with one variable, Linear regression with multiple variables, Logistic Regression,

Polynomial regression, Classification: Decision trees, Naive Bayes classifier, Support Vector Machine, Kernel functions, Clustering: K-mean, hierarchical.

Unit-III **(10 hours)**

Bias-variance tradeoff, Feature scaling, feature selection methods - Lasso, Ridge, Curse of dimensionality, dimensionality reduction methods - Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA).

Unit IV **(10 hours)**

Introduction, Activation Artificial Neural Networks: Function, Optimization algorithm, Gradient descent, Backpropagation Algorithms, Training Procedures Networks- Perceptron's, Multilayer Perceptron's, Hopfield neural network, recurrent neural network, Self-organising maps, Generative AI.

Readings:

1. Alpaydin, Ethem. *Introduction to machine learning*. MIT Press, 2020.

2. Russell, Stuart J., and Peter Norvig. *Artificial intelligence: a modern approach*. Pearson, 2016.

3. Bishop, Christopher M., and Nasser M. Nasrabadi. *Pattern recognition and machine learning*. Vol. 4. Springer, 2006.

4. Shalev-Shwartz, Shai, and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge University Press, 2014.

5. Michalski, Ryszard Stanislaw, Jaime Guillermo Carbonell, and Tom M. Mitchell, eds. *Machine learning: An artificial intelligence approach*. Springer Science & Business Media, 2013.

List of Practicals:

1. Write a program to implement the working of Breadth-First Search (BFS) and Depth-First Search (DFS). **(2 hours)**
2. Build a simple Linear Regression model with one and multiple variables using scikit-learn. **(2 hours)**
3. Implement Logistic Regression for binary classification (e.g., spam detection). **(2 hours)**
4. Train and visualise Decision Tree and Naive Bayes classifiers on a sample dataset. **(4 hours)**
5. Build and evaluate a Support Vector Machine (SVM) with the kernel trick on a classification problem. **(4 hours)**
6. Implement K-Means and Hierarchical clustering and visualise clustering results. **(4 hours)**
7. Apply feature scaling and evaluate its impact on model performance. Further, use Lasso and Ridge regression to perform feature selection and regularisation, and compare model performance with and without scaling and regularisation. **(4 hours)**
8. Perform Dimensionality Reduction using PCA and LDA and visualise transformed features. **(4 hours)**
9. Build a Multilayer Perceptron from scratch or using TensorFlow/Keras; train it on a classification task (e.g., MNIST). **(4 hours)**

SBC201: SCIENTIFIC WRITING AND COMPUTATIONAL ANALYSIS TOOLS [0-0-2]

Course Code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice	
SBC201	2	0	0	2	Graduation

Course Objective:

The course aims to equip students with the skills to create professional documents using LaTeX and master MATLAB for data analysis, signal processing, image processing, and system modelling. Students will also learn to work with MATLAB functions, scripts, and Simulink for simulation-based tasks.

Course Learning Outcomes:

Upon successful completion of this course, students will be able to

1. Create a basic LaTeX document, format text with different styles, and apply fundamental document structures such as sections, paragraphs, and formatting options.
2. create and manipulate matrices and arrays in MATLAB, understanding their importance in numerical computation and data analysis.
3. apply optimisation techniques and perform curve fitting tasks in MATLAB, including fitting data to a model and minimising error.

Syllabus:

Unit-I

LaTex : Basic Document Setup, and Formatting Text, Lists and Tables, Multi-Column Layouts, Mathematical Equations, Figures and Graphics, Customizing Fonts and Colors, References, Citations, and Headers/Footers/Page Numbering, Creating a Resume or CV.

Unit-II

MATLAB: Basic MATLAB Operations, Matrices and Arrays, Plotting and Visualization, Loops and Conditional Statements, Functions and Scripts, File Handling, Signal Processing, Image Processing, Optimization and Curve Fitting, Simulink Basics

Readings:

1. https://guides.nyu.edu/LaTeX/sample-document
2. Brian R. Hunt, Ronald L. Lipsman, Jonathan M. Rosenberg. *A Guide to MATLAB: For Beginners and Experienced Users*, Cambridge University Press, 2017.

List of Practicals :

1. Basic Document Setup and Formatting Text: Create a new document with a specific page size, orientation, and margins, Format a paragraph using different font styles (bold, italic, underline) and sizes, Apply alignment options (left, right, center, justified) to different text blocks, Use styles to format headings and body text consistently. **(4 hours)**

2. Lists and Tables, Multi-Column Layouts: Create a bulleted and numbered list for an agenda or shopping list, design a table to represent student grades or inventory details, merge and split cells in the table, and apply borders and shading. Create a newsletter-style document using two or three-column layouts. **(4 hours)**

3. Mathematical Equations: Insert and format basic math expressions, such as fractions, superscripts, and subscripts. Write complex equations, like the quadratic formula, matrices, and integrals, using equation tools or LaTeX syntax (if applicable), and align multiple equations properly using equation editors or tab alignment. **(4 hours)**

4. Figures and Graphics, Customising Fonts and Colours: Insert an image or figure into the document with a caption, Resize and position the image using wrap text and alignment options, Change font family and colour scheme for different document elements, Create a cover page with custom fonts, colours, and images. **(4 hours)**

5. References, Citations, and Headers/Footers/Page Numbering: Insert a bibliography and add citations using a reference manager or built-in citation tools, Apply consistent header and footer designs across the document, Insert and format page numbers (e.g., Roman numerals for intro, Arabic for content), Create a Table of Contents and update it automatically. **(4 hours)**

6. Creating a Resume or CV: Use a template or design a CV layout from scratch with appropriate sections (Education, Experience, Skills), Insert a profile photo, format contact details, and add social media links, Apply proper use of whitespace, bullets, and alignment to ensure clarity and readability, Export the CV as a PDF and check formatting consistency. **(4 hours)**

7. Introduction to MATLAB Interface and Basic Commands: Using command window, editor, and workspace, Arithmetic operations, using help, clc, clear, who, whos. Variable Assignment and Data Types: Creating scalars, vectors, complex numbers, Type conversion, and precision handling, Matrix Creation and Manipulation: Defining matrices, transposition, reshaping, Indexing, slicing, concatenation, Matrix multiplication, inversion, determinant, eigenvalues. **(4 hours)**

8. Plotting and Visualization 2D and 3D Plotting: plot(), subplot(), title(), xlabel(), ylabel(), legend(), 3D plots: mesh(), surf(), contour(), Data Visualization with Customization: Bar graphs, pie charts, histograms, Line styles, markers, color changes. **(4 hours)**

9. Loops and Conditional Statements: Using for, while, and if-else, writing loops to sum a series, the Fibonacci series, and Conditionals to check prime numbers or grading systems, Functions and Scripts, Creating User-Defined Functions, writing a function to compute factorial, and standard deviation. **(4 hours)**

10. Using Scripts for Automation: Writing scripts to read input, process, and display output, File Handling: Reading and Writing Files, reading data from .txt, .csv using fopen, fscanf, textscan, readmatrix(), Writing output to files. **(4 hours)**

11. Signal Processing: Basic Signal Generation and Analysis, generating sine, square, and triangular waves, plotting signals, and performing FFT, Filtering Signals: Applying FIR and IIR filters, using filter (), butter (), and freqz(). **(4 hours)**

12. Image Processing: Image Reading and Display, Read and display an image using imread(), imshow(), and rgb2gray(), Image Enhancement and Filtering: Histogram equalization, edge detection, smoothing. **(8 hours)**

13. Optimisation and Curve Fitting: Curve Fitting using polyfit() and fit(), Fit a polynomial or custom function to data, Evaluate goodness of fit, optimisation using fminsearch, fmincon: Minimise a nonlinear function with constraints. **(8 hours)**

List of DSEs for Semester II

DSE201: Social Networks Analysis [3-0-1]

Course title & Code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice	
DSE 201	4	3	0	1	Basic knowledge of graphs

Course Objectives:

The course aims to equip students with various SNA approaches to data collection, cleaning, and pre-processing of network data.

Course Learning Outcomes:

On completing this course, the student will be able to:

1. explain the basic concepts and principles of social network.
2. identify different types of social networks and their characteristics.
3. implement and apply various social network analysis techniques, such as influence maximisation, community detection, link prediction, and information diffusion.
4. apply network models to understand phenomena such as social influence, diffusion of innovations, and community formation.

Syllabus:

Unit-I **(9 hours)**

Introduction to Social Network Analysis, Types of Networks, Nodes, Edges, Node Centrality, betweenness, closeness, eigenvector centrality, network centralisation, Assortativity, Transitivity, Reciprocity, Similarity, Degeneracy and Network Measure, Networks Structures, Network Visualisation, Tie Strength, Trust, Understanding Structure Through User Attributes and Behaviour.

Unit-II **(12 hours)**

Link Analysis and Link Prediction: Applications of Link Analysis, Signed Networks, Strong and Weak Ties, Link Analysis and Algorithms, Page Rank, Personalized PageRank, DivRank, SimRank, PathSim. Temporal Changes in a Network, Evaluation Link Prediction Algorithms, Heuristic Models, Probabilistic Models, Applications of Link Prediction.

Unit-III **(12 hours)**

Community Detection: Applications of Community Detection, Types of Communities, Community Detection Algorithms, Disjoint Community Detection, Overlapping Community Detection, Local Community Detection, Evaluation of Community Detection Algorithms.

Unit IV: **(12 hours)**

Influence Maximisation: Applications of Influence Maximisation, Diffusion Models, Independent Cascade Model, Linear Threshold Model, Triggering Model, Time-Aware Diffusion Model, Non-Progressive Diffusion Model. Influence Maximization Algorithms, Simulation-Based Algorithms, Proxy-Based Algorithms, Sketch-Based Algorithms, Community-Based Influence Maximization, and Context-Aware Influence Maximization.

Readings:

1. Chakraborty, T. *Social Network Analysis*, Wiley India, 2021.
2. Knoke, D. and Yang, S. *Social network analysis*. SAGE Publications, 2019.
3. Golbeck, J. *Analyzing the social web*, Morgan Kaufmann, 2013.
4. Wasserman, S and Faust, K. *Social network analysis: Methods and applications*, Cambridge University Press, 2012.
5. Newman, M.E.J. *Networks: An introduction*. Oxford University Press, 2010.
6. Chen, W. Castillo, C and Lakshmanan, L.V.S. *Information and influence propagation in social networks,* Springer Nature, 2014
7. Srinivas, V and Mitra, P. *Link prediction in social networks: role of power law distribution,* New York: Springer International Publishing, 2016

List of Practicals :

1. Implement heuristic link prediction methods such as Common Neighbors, Adamic-Adar, and Jaccard Coefficient to identify potential future links in a network. **(4 hours)**

2. Apply SimRank and PathSim to compute similarity scores for link prediction in graph-based data. **(4 hours)**

4. Compare heuristic and probabilistic link prediction models using evaluation metrics like precision and recall. **(4 hours)**

5. Use the Girvan–Newman algorithm to detect disjoint communities by iteratively removing high-betweenness edges. **(4 hours)**

6. Apply the Louvain or Leiden algorithm to uncover modular communities in large-scale networks. **(4 hours)**

7. Evaluate community detection results using modularity and normalised mutual information (NMI). **(4 hours)**

8. Simulate the Independent Cascade Model (ICM) to observe the spread of influence from

a given set of seed nodes. **(4 hours)**

9. Simulate the Linear Threshold Model (LTM) to model influence propagation based on node activation thresholds. **(4 hours)**
10. Implement Greedy and CELF algorithms to efficiently select the top-k influential nodes for maximizing spread. **(4 hours)**
11. Benchmark different influence maximisation techniques, including simulation-based, sketch-based, and community-based methods. **(4 hours)**

DSE202: Combinatorial Optimization [3-0-1]

Course title & Code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice	
DSE202	4	3	0	1	knowledge of Linear Algebra

Course Objectives:

The course aims to equip students with the technique of linear and integer programs to solve optimization problems via LP-based solutions to shortest path problems, minimum spanning tree problem, max-flow problem and maximum matching problem.

Course Learning Outcomes (CO):

On completion of this course, the student will be able to:

1. model problems using linear and integer programs.
2. differentiate between the computational complexities of LP and IP.
3. understand polyhedral analysis and apply it to develop algorithms.
4. understand the concept of duality and use it to design exact and approximate algorithms.
5. understand and explain the mathematical theory forming the basis of many algorithms for combinatorial optimization (particularly graph theoretic).

Syllabus:

Unit-I **(15 hours)**

Introduction: Optimization problems, neighborhoods, local and global optima, convex sets and functions, simplex method, degeneracy; duality and dual simplex algorithm, computational considerations for the simplex and dual simplex algorithms-Dantzig-Wolfe algorithms.

Unit-II **(10 hours)**

Integer Linear Programming: Cutting plane algorithms, branch and bound technique and

approximation algorithms for traveling salesman problem.

Unit-III **(15 hours)**

Graph Algorithms: Primal-Dual algorithm and its application to shortest path (Dijkstra's algorithm, Floyd-Warshall algorithms), max-flow problem (Ford and Fulkerson labeling algorithms), matching problems (bipartite matching algorithm, non-bipartite matching algorithms, bipartite weighted matching-hungarian method for the assignment problem, non-bipartite weighted matching problem), efficient spanning tree algorithms.

Unit-IV **(5hours)**

Matroids: Independence Systems and Matroids, Duality, Matroid Intersection.

Readings:

1. Korte, Bernhard H., Jens Vygen, B. Korte, and J. Vygen. *Combinatorial optimization*. Springer, 2018.
2. Matoušek, Jiří, and Bernd Gärtner. *Understanding and using linear programming*. Springer, 2007.
3. Papadimitriou, Christos H., and Kenneth Steiglitz. *Combinatorial optimization: algorithms and complexity*. Dover Publication, 1998.
4. Bazaraa, Mokhtar S., John J. Jarvis, and Hanif D. Sherali. *Linear programming and network flows*. John Wiley & Sons, 2011.
5. Taha, Hamdy A. *Operations research: an introduction*. Pearson Education India, 2014.

List of Practicals:

1. WAP to implement Simplex Method (**4 hours)**
2. WAP to implement Dual Simplex Method (**6 hours)**
3. WAP to implement Primal-Dual algorithm for shortest path problem **(6 hours)**
4. WAP to implement Floyd-Warshall algorithm for shortest path problem (**6 hours)**
5. Implement algorithms for matching problems. **(8hours)**

DSE203: Cyber Security [3-0-1]

Course title & Code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice	
DSE203	4	3	0	1	Graduation

Course Objectives:

This course will be responsible for laying the foundation for creating a comprehensive understanding and expertise in the field of cybersecurity. This paper will set the level field for all the students to be able to come on par and move together as they must go deeper into hardcore cybersecurity topics during the course duration.

Course Learning Outcomes:

On completion of this course, the student will be able to:

1. state the need and scope for cyber laws.
2. enumerate various network attacks, describe their sources, and mechanisms of prevention.
3. describe the genesis of SCADA policies and their implementation framework.
4. carry out malware analysis.

Syllabus:

Unit-I **(7 hours)**

Introduction: Cyberspace, Internet, Internet of things, Cyber Crimes, cyber criminals, Cyber security, Cyber Security Threats, Cyber laws and legislation, Law Enforcement Roles and Responses.

Unit-II **(15 hours)**

Cyberspace Attacks: Network Threat Vectors, MITM, OWASP, ARP Spoofing, IP & MAC Spoofing, DNS Attacks, SYN Flooding attacks, UDP ping-pong and fraggle attacks, TCP port scanning and reflection attacks, DoS, DDOS. Network Penetration Testing Threat assessment, Penetration testing tools, Penetration testing, Vulnerability Analysis, Threat matrices, Firewall and IDS/IPS, Wireless networks, Wireless Fidelity (Wi-Fi), Wireless network security protocols, Nmap, Network fingerprinting, BackTrack, Metasploit.

Unit-III: **(15 hours)**

Introduction to SCADA (supervisory control and data acquisition): Understanding SCADA security policies, SCADA Physical and Logical Security, understanding differences between physical and logical security, define perimeter controls and terms, define various security zones, understand communication cyber threats, Understand firewall, architectures.

Unit-IV **(8 hours)**

Introduction Malware Analysis: Static Analysis, Code Review, Dynamic Analysis, Behavioral analysis of malicious executable, Sandbox Technologies, Reverse-engineering malware, Defeat anti-reverse engineering technique, automated analysis, intercepting network connections, Network flow analysis, Malicious Code Analysis, Network analysis.

Readings:

1. Peter W. Singer and Allan Friedman, *Cybersecurity and Cyberwar*, Oxford University Press, 2014.
2. Michael Sikorski, Andrew Honig, *Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software* 2012, No Starch Press, San Francisco.
3. Dejay, Murugan, *Cyber Forensics* Oxford university press India Edition, 2018.
4. R. Rajkumar, D. de. Niz and M. Klein, *Cyber Physical Systems*, Addision-Wesely, 2017
5. Jonathan Clough, *Principles of Cybercrime*, Cambridge University Press, 24-Sep-2015.

References:

1. CEH official Certified Ethical Hacking Review Guide, Wiley India Edition, 2015.

List of Practicals:

Disclaimer: Please note that these lab exercises are for educational purposes only. All activities should be performed within the boundaries of the law.

1. Identify your machine's IP configuration using *ipconfig (Windows)* or i*fconfig (Linux)*. Ping various local and remote hosts to verify connectivity, trace the route to a website using *tracert* or *traceroute,* perform DNS lookups using *nslookup*, and gather domain registration details using the *whois* tool. **(2 hours)**
2. Use tools like *John the Ripper* and *Hydra* to crack passwords. The task will involve using wordlist-based and brute-force techniques to break into password-protected accounts or services. **(4 hours)**

3. SET tool is one of the tools available in Kali Linux. Explore the tool and list down all types of Social Engineering attacks available in the tool. Further, make a document mentioning the Name of the attack, and how the attack can be launched. How can I make my system safe from these attacks? **(6 hours)**
4. Use the Social Engineering Toolkit (SET) available in Kali Linux to clone a vulnerable website (e.g., http://www.itsecgames.com/). Compare the cloned website with the original, explore different social engineering attacks such as phishing or credential harvesting. **(4 hours)**
5. Use *theHarvester* tool to collect at least 25 email addresses from various publicly available sources on the internet. These addresses can later be used for phishing simulations or further social engineering attacks. **(2 hours)**
6. Utilize tools from the *iHunt intelligent framework* available on web to create a sock puppet account. Document the tools used, their functionality, and how this can be applied in a simulated information-gathering exercise. **(2 hours)**
7. Using the email addresses collected in Lab 6, perform further information gathering. Use search engines and social media platforms to find publicly available information about the email owners. Document the results using the tools given on the *iHunt intelligent framework* for email investigation. **(2 hours)**
Note: Don't actually contact the person of interest. Just gather information which available publicly. Don't break any law.
8. Conduct a vulnerability scan on target systems using *NMap*. Analyze the results to identify potential vulnerabilities such as open ports, weak configurations, and security loopholes. **(4hours)**
9. Use the *Metasploit framework* to exploit vulnerabilities and gain access to a remote Linux machine. The exercise involves using Kali Linux with Metasploit to exploit a vulnerable system.**(4hours)**
10. A lab exercise related to SCADA & Industrial Control Systems (ICS) Security **(4 hours)**

DSE204: Information Retrieval [3-0-1]

Course Code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice	
DSE204	**4**	**3**	**0**	**1**	**Basic understanding of Statistics and Probability is required.**

Course Objectives:

This course introduces the basics of Information Retrieval (IR), focusing on how information is organized, searched, and ranked. Students will learn about different search models, document processing techniques, and ways to measure search accuracy. The course also covers web search methods, including link analysis and web crawling.

Course Learning Outcomes:

Upon successful completion of this course, students will be able to

1. exhibit the understanding of the fundamental concepts of Information Retrieval (IR), including information need, relevance, and early developments in IR systems.
2. apply different retrieval models, such as Boolean retrieval and ranked retrieval, to effectively search and organize information.
3. evaluate search performance using precision-recall, ranking measures, and other standard evaluation metrics.
4. process and represent documents using techniques like vector space modelling, feature selection, stemming, and similarity measures.
5. explore web search techniques, including link analysis methods like PageRank and HITS, as well as web crawling strategies.

Syllabus:

Unit-I **(15 hours)**

Introduction to Information Retrieval (IR), information, information need, and relevance, the IR system and its components, early developments in IR, user interfaces in IR, retrieval and IR models, Boolean retrieval, term vocabulary and postings list, ranked retrieval, inverted index, index construction, index compression.

Unit-II **(10 hours)**

Document processing, document representation techniques, vector space model, feature selection for IR, stop words, stemming, concept of document similarity, evaluation of information retrieval systems, notion of precision and recall, precision-recall curve.

Unit-III **(10 hours)**

Standard performance measures, MAP, reciprocal ranks, F-measure, NDCG, rank correlation, standard datasets for IR evaluation, web search and link analysis, web crawling techniques, link analysis methods, PageRank algorithm, HITS algorithm.

Unit-IV **(10 hours)**

Classification and Clustering: Notion of supervised and unsupervised algorithms, Naive Bayes, nearest neighbour and Rochio's algorithms for text classification, text clustering methods such as K-Means.

<u>Readings:</u>

1. Baeza-Yates, Ricardo, and Berthier Ribeiro-Neto. *Modern information retrieval*. Vol. 463, no. 1999. New York: ACM Press, 1999.
2. Schütze, Hinrich, Christopher D. Manning, and Prabhakar Raghavan. *Introduction to information retrieval*. Vol. 39. Cambridge: Cambridge University Press, 2008.
3. Grossman, David A., and Ophir Frieder. *Information retrieval: Algorithms and heuristics*. Vol. 15. Springer Science & Business Media, 2004.
4. Buttcher, Stefan, Charles LA Clarke, and Gordon V. Cormack. *Information retrieval: Implementing and evaluating search engines*. Mit Press, 2016.
5. Croft, W. Bruce, Donald Metzler, and Trevor Strohman. *Search engines: Information retrieval in practice*. Vol. 520. Reading: Addison-Wesley, 2010.

<u>List of Practicals</u>

1. Design and implement a Boolean retrieval system that can process queries using AND, OR, and NOT operators on a given set of text documents. **(2 hours)**
2. Develop a text preprocessing pipeline that performs tokenization, case folding, stop-word removal, and stemming on a document corpus. **(2 hours)**
3. Write a program that constructs an inverted index for a collection of documents and supports retrieval of documents based on single-word queries, then implement index compression using variable-byte encoding and analyze the reduction in index size and its effect on retrieval performance. **(4 hours)**
4. Compute the Term Frequency-Inverse Document Frequency (TF-IDF) values for each term in a document collection and store them in a structured format. **(2 hours)**
6. Implement a ranked retrieval system using the vector space model and cosine similarity to score and rank documents based on a user query. **(2 hours)**
7. Use a publicly available IR dataset to implement and compare different retrieval models (e.g., Boolean, Vector Space), then calculate evaluation metrics including precision, recall, F1-score, MAP, and NDCG using provided relevance judgments, and finally, generate precision-recall curves for multiple queries to interpret the performance of each model. **(4 hours)**
8. Implement both the PageRank and HITS algorithms on a manually created web graph, simulate multiple iterations to observe rank convergence, and compute hub and authority scores for each node. **(4 hours)**
9. Train and evaluate a Naive Bayes classifier to categorize documents into predefined classes such as spam vs. ham or news categories. **(2 hours)**
10. Apply the K-Means clustering algorithm to a set of TF-IDF document vectors and visualize the resulting clusters using a 2D projection. **(4 hours)**
11. Develop a basic web crawler that extracts and stores titles and content from publicly accessible websites while respecting robots.txt. **(4 hours)**

DSE205: Digital Image Processing [3-0-1]

Course title & Code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice	
DSE205	4	3	0	1	Graduation

Course Objectives:

The course will thoroughly introduce image enhancement in the spatial and frequency domains, followed by image morphological operations such as dilation, erosion, hit-or-miss transformations, image segmentation and image compression.

Course Learning Outcomes:

Upon successful completion of this course, the student will

1. be able to enhance the quality of an image using various transformations.
2. analyze the transform of an image in spatial domain to frequency domain.
3. apply required morphological operations to an image.
4. adequate to segment an image using various approaches.

Syllabus:

Unit-I **(7 Hours)**
Introduction: Applications of digital image processing, steps in digital image processing: image acquisition, image sampling and quantization, basic relationships between pixels.

Unit-II **(14 Hours)**
Image enhancement: grey level transformations, histogram processing, local enhancement, image subtraction, image averaging, spatial filtering: smoothing and sharpening filters, Discrete Fourier transformation, filtering in the frequency domain: smoothing and sharpening filters, image restoration in spatial and frequency domains.

Unit-III **(12 Hours)**
Morphological image processing and Image Compression: erosion and dilation, opening and closing, hit-or-miss transformation, and some basic morphological algorithms, Image compression models, error-free compression techniques, lossy compression techniques, JPEG, MPEG.

Unit-IV **(12 Hours)**
Image segmentation Point: line and edge detection, gradient operator, edge linking and boundary detection, thresholding, region-based segmentation, representation schemes like chain codes, polygonal approximations, boundary segments, skeleton of a region, boundary descriptor, regional descriptor.

Readings:

1. Gonzalez, Rafael C. and Woods, Richard E., *Digital Image Processing*, 4th edition, Pearson Education, 2018.
2. Annadurai, S. and Shanmugalakshmi, R., *Fundamentals of Digital Image Processing*, 1st edition, Pearson, 2006.
3. Joshi, M. A., *Digital Image Processing: An Algorithmic Approach*, 2nd edition, PHI Learning, 2020.

References:

1. Jahne, Bernd, *Digital Image Processing*, 6th edition, Springer, 2005.
2. Chandra, B. and Majumder, D.D., *Digital Image Processing and Analysis*, 2nd edition, Prentice Hall India Learning Private Limited, 2011.

List of practicals:

You may choose a suitable programming language and all necessary/relevant inputs/ problems.

1. Implement a program for displaying grey-scale and RGB colour images, with those for accessing pixel locations, and investigate adding and subtracting a scalar value from an individual location. **(2 Hours)**

2. Implement a program to perform the image sampling and quantization technique and display the resulting images. **(4 Hours)**

3. Implement a program to apply histogram equalization to a colour image, and apply contrast stretching to the colour example image. Experiment with different parameter values to find an optimum for the visualization of this image. **(2 Hours)**

4. Implement a program to add different levels of salt and pepper and Gaussian noise to images, both in colour and grey-scale. Investigate the usefulness of all filtering for removing different levels of image noise. **(2 Hours)**

5. Write a program for all image restoration techniques. **(4 Hours)**

6. Write a program for the Fourier transform of an image. **(2 Hours)**

7. Write a program for image spatial and sharpening filtering. **(4 Hours)**

8. Implement a program to find the result of erosion, dilation, opening and closing with the below structuring element and choose any image. **(2 Hours)**

a)	1 0 0	**b)** 1 1 1
	0 1 0	1 1 1
	0 0 1	1 1 1

9. Write a program for all the compression techniques of an image. **(4 Hours)**

10. Write a program for all image segmentation techniques. **(4 Hours)**

DSE206: DATA MINING [3-0-1]

Course title & Code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/Practice	
DSE 206	4	3	0	1	Graduation

Course Objectives:

In this course, the objective is to introduce the KDD process. The course should enable students to translate real-world problems into predictive and descriptive tasks. The course also covers data cleaning and visualization, supervised and unsupervised mining techniques.

Course Learning Outcomes:

At the end of the course, the student will be able to:

1. distinguish between the process of knowledge discovery and Data Mining.
2. play with basic data exploration methods to develop understanding of given data.
3. identify suitable pre-processing method for give problem.
4. describe different data mining tasks and algorithms.
5. use programming tools (e.g. Weka/Python/R etc) for solving data mining tasks.
6. follow formal notations and understand the mathematical concepts underlying data mining algorithms

Syllabus:

Unit I **(9 hours)**

Overview: The process of knowledge discovery in databases, predictive and descriptive data mining techniques, and unsupervised learning techniques. Data pre-processing: Data cleaning, Data transformation, Data reduction, Discretization

Unit-II **(12 hours)**

Classification: Supervised learning/mining tasks, Decision trees, Decision rules, Statistical (Bayesian) classification, Instance-based methods (nearest neighbour), Evaluation and Validation methods.

Unit-III **(12 hours)**

Clustering: Basic issues in clustering, Partitioning methods (k-means, expectation maximisation), Hierarchical methods for clustering, Density-based methods, Cluster Validation methods and metrics.

Unit-IV **(12 hours)**

Association Rule Mining: Frequent item set, Maximal and Closed item sets, Apriori property, Apriori algorithm.

Readings:

1. Han, Jiawei, Jian Pei, and Hanghang Tong. *Data Mining: Concepts and Techniques* 3nd edition. Morgan Kaufmann, 2022.
2. Zaki, Mohammed J., WM Jr, *Data Mining and Analysis: Fundamental Concepts and Algorithms,* Cambridge University Press, 2014.
3. Tan, Pang-Ning, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining*, Addison Wesley, 2006.
4. Charu, C. *Data Mining: The Textbook*, Springer, 2015

<u>List of Practicals:</u>

1. Data Preprocessing & Visualization (4 Hours)

- Load and explore real-world datasets using Python/Weka/R
- Perform data cleaning: handle missing values, detect/treat outliers
- Apply data transformation: normalization, encoding categorical variables
- Use data reduction techniques: PCA, sampling
- Apply discretization/binning on continuous features
- Visualize data using histograms, boxplots, scatter plots, and heatmaps

2. Supervised Learning – Classification (4 Hours)

- Implement a Decision Tree classifier and visualise the tree
- Apply Naive Bayes classifier and evaluate accuracy
- Implement k-Nearest Neighbor (k-NN) classifier with different values of *k*
- Evaluate classifiers using confusion matrix, precision, recall, and F1-score
- Perform k-fold cross-validation and plot ROC-AUC curves

3. Unsupervised Learning – Clustering (8 Hours)

- Apply k-Means clustering and determine optimal k using Elbow/Silhouette method
- Perform Agglomerative Hierarchical clustering and plot dendrogram
- Apply DBSCAN for density-based clustering and tune parameters (eps, minPts)
- Visualize and interpret clustering results

4. Association Rule Mining (4 Hours)

- Use Apriori algorithm to find frequent itemsets from transaction data
- Generate association rules using confidence and lift thresholds
- Analyze discovered rules for business decision insights

5. Mini-Project (10 Hours)

- Select a real-world dataset (e.g., healthcare, e-commerce, finance)
- Apply full KDD process: preprocessing, modelling, (classification/clustering/association), and evaluation
- Summarize findings with visualizations and model insights

Annexure-3

Master of Operational Research

Two Year Programme

(PG Curriculum Framework 2024 based on NEP 2020)

Academic Session 2025-26

Department of Operational Research
Faculty of Mathematical Sciences
University of Delhi
Delhi – 110007

Index

PG Curriculum Framework 2024 based on NEP 2020

Course Structure: Sem-I to Sem-IV, Structure-1 & Structure-2 of Two-Year PG Program in Operational Research, PGCF

1st year of PG curriculum structure for 2-year PG Program

Semester	DSC (12 Credits)	DSE 1, DSE 2 (8 credits) OR DSE 1 & GE 1 (8 credits)	Skill-based Course (2-credit course)	Dissertation/Academic Project/Entrepreneurship	Total Credits	One GE to be offered by the Department
	Credit Distribution: **(4 * 3) = 12**	**Credit Distribution:** **(4 * 2) = 8**	**Credit Distribution:** **(2*1) = 2**		**(12+8+2)** **= 22**	
Sem I	DSC-1: Inventory Management DSC-2: Linear Programming & Extensions DSC-3: Statistics	Any **two** of the following: DSE-1(a): Mathematics for OR DSE-1(b): Decision Theory DSE-1(c): Design Thinking and Innovation DSE-1(d): Game Theory with Behavioral Aspects DSE-1(e): Simulation Modeling DSE-1(f): Software Engineering	Database Management System	Nil		Any **one** of the following: GE-1(a): Inventory Management GE-1(b): Linear Programming & Extensions GE-1(c): Game Theory with Behavioral Aspects GE-1(d): Simulation Modeling

Sem II	DSC-4: Optimization Techniques DSC-5: Queueing Theory DSC-6: Python Programming for Decision-Making	Any **two** of the following: DSE-2(a): Applied Multivariate Analysis DSE-2(b): Financial Management DSE-2(c): Fundamentals of Managerial Economics DSE-2(d): Marketing Research DSE-2(e): Quality Management DSE-2(f): Soft Computing	Spreadsheet and Data Visualization	Nil	22	Any **one** of the following: GE-2(a): Queueing Theory GE-2(b): Marketing Research GE-2(c): Quality Management GE-2(d): Soft Computing

Structure 1 (Level 6.5): PG Curriculum Structure with only course work

Semester	DSC (8 credits)	DSE (12 credits)	Skill-based course/workshop/Specialized Laboratory/Internship/Apprenticeship/Hands- on Learning (2credits)	Dissertation/Academic Project/Entrepreneurship	Total Credits
Sem III	DSC-7: Econometric Modeling & Forecasting DSC-8: Marketing Management	*	**	NIL	22
Sem IV	DSC-9: Reliability & Maintenance Theory DSC-10: Scheduling Techniques	*	**	NIL	22

Structure 2 (Level 6.5): PG Curriculum Structure with only course work + Research

Semester	DSC (8 credits)	DSE (8 credits)	Skill-based course/workshop/Specialized Laboratory/Internship/Apprenticeship/Hands-on Learning (2credits)	Dissertation/Academic Project/Entrepreneurship	Total Credits
Sem III	DSC-7: Econometric Modeling & Forecasting DSC-8: Marketing Management	*	NIL	(6 credits)	22
Sem IV	DSC-9: Reliability & Maintenance Theory DSC-10: Scheduling Techniques	*	NIL	(6 credits)	22

*** POOL OF TENTATIVE DSE COURSES IN SEMESTER-III & SEMESTER- IV, STRUCTURE 1 AND STRUCTURE 2**

S. No.	Title of the paper
i.	Health Care Management
ii.	Revenue Management
iii.	Supply Chain Management
iv.	Advanced Inventory Management
v.	Queueing Networks
vi.	Multicriteria Decision-Making Techniques
vii	Data Warehousing and Data Mining
viii.	Dynamic Optimization
ix.	Portfolio Optimization
x.	Stochastic Modeling
xi.	Advanced Marketing Management
xii.	Social Media Analytics
xiii.	Marketing Analytics
xiv.	Text Analytics for Management
xv.	Pattern Recognition
xvi.	Deep Learning
xvii.	Advanced Software Reliability Models
xviii.	Fuzzy Sets & Logic
xix.	Numerical Optimization

xx.	Logistics and Network Optimization
xxi.	Financial Modeling
xxii.	Warranty Modeling & Analysis
xxiii.	Bayesian Forecasting
xxiv.	Reliability Testing & Prediction
xxv.	Prognostics & Health Management of Systems
xxvi.	Bayesian Reliability
xxvii.	Design & Analysis of Experiments
xxvii.	OR for Public Policy
Any other paper would also be added to this pool if need be.	

** Skill-based course/workshop/Specialized Laboratory/Internship/Apprenticeship/Hands-on Learning (2credits): To be decided later.
*** GEs to be offered by the Department in Semester-III & Semester-IV(4 credits): To be decided later.

Based on Postgraduate Curriculum Framework 2024 based on NEP 2020

MASTER OF OPERATIONAL RESEARCH
DEPARTMENT OF OPERATIONAL RESEARCH
UNIVERSITY OF DELHI

POSTGRADUATE PROGRAM OF STUDY

SYLLABI OF SEMESTERS I & II

Discipline Specific Core - Semester I

DSC - 1: Inventory Management

DSC - 2: Linear Programming & Extensions

DSC - 3: Statistics

DISCIPLINE SPECIFIC CORE
DSC-1: INVENTORY MANAGEMENT

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Prerequisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Inventory Management (DSC-1)	**4**	**3**	**1**	**0**	**-**	Graduation

Learning Objectives:

The learning objectives of this course are as follows:
- To impart knowledge of the role of inventory in organizations and core concepts of inventory management with an understanding of fundamental inventory control procedures and their usage.
- To provide students with a thorough understanding of basic inventory models and a rigorous mathematical framework to develop mathematical models to analyze and optimize inventory systems.
- To provide the students with a comprehensive study of various application areas of inventory models through case studies and relevant examples.

Learning Outcomes:

Students completing this course will be able to:
- Identify the goals and objectives of inventory management and describe the importance of stocks in an organization and the reasons for holding stock.
- Explain the various costs related to the inventory system.
- Understand the various selective inventory control techniques and their applications.
- Capability to develop deterministic inventory models: economic order quantity and its extensions, All units and incremental quantity discounts models, Joint and Individual order policies, and Production scheduling models.
- Understand and develop stochastic inventory models and setting safety stocks. Apply and extend inventory models to analyse real-world systems.

Syllabus of DSC-1:

Unit I: Introduction to Inventory Systems (9 hours)
Analytical structure of Production and Inventory problems. Objectives of Inventory Management. Factors influencing inventories. Inventory-related costs. Properties of Inventory systems. Selective Inventory control techniques and their applications. Concept of Lead Time. Introduction to Just in Time (JIT) and Vendor Managed Inventory (VMI).

Unit II: Deterministic Inventory Models (15 hours)
Deterministic inventory models, economic order quantity and its extensions: without and with lead time. Finite replenishment rate Inventory models without and with planned shortages. Inventory models with partial backlogging and lost sales. Discrete Demand Model. Multi-item Inventory models with constraints. Quantity discounts: All units and incremental. Joint and Individual Ordering Policies.

Unit III: Production Planning Models **(9 hours)**
Aggregate Production Planning Models: Fixed workforce model. Variable workforce model. Dynamic lot size models: Wagner-Whitin Algorithm, Silver-Meal heuristic.

Unit IV: Stochastic Inventory Models **(12 hours)**
Stochastic Inventory models, Newsvendor model and its extensions: Instantaneous and uniform demand with discrete and continuous cases; without and with lead time. Transformations for the equivalence of instantaneous and uniform demand models. Power demand pattern inventory model. Periodic review models. Safety stocks, Service levels and reorder level.

Tutorial component (if any) - Yes **(15 hours)**

Practical component (if any) - Nil

Essential Readings:

1. Axsäter, S. (2015). *Inventory Control.* (Third Edition). Springer.
2. Hadley, G., & Whitin, T. M. (1963). *Analysis of Inventory Systems*. Prentice-Hall.
3. Johnson, L.A., & Montgomery, D.C. (1974) *Operations Research in Production Planning, Scheduling and Inventory Control*. Wiley, New York.
4. Muckstadt, J. A., & Sapra, A. (2010). *Principles of Inventory Management: When You Are Down to Four, Order More*. Springer Science & Business Media.
5. Naddor, E. (1966). *Inventory Systems*. Wiley.
6. Silver, E. A., Pyke, D. F., & Peterson, R. (1998). *Inventory Management and Production Planning and Scheduling*. (Third Edition). Wiley.
7. Waters, D. (2008). *Inventory Control and Management.* (Second Edition). John Wiley & Sons.
8. Zipkin, H. P. (2000). *Foundations of Inventory Management*. McGraw-Hill.

Suggested Readings: Nil

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

DISCIPLINE SPECIFIC CORE
DSC-2: LINEAR PROGRAMMING AND EXTENSIONS

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Linear Programming and Extensions (DSC-2)	**4**	**3**	**1**	**0**	**-**	Graduation

Learning Objectives:

The learning objectives of this course are as follows:
- To impart knowledge of formulating practical problems using the linear programming method and its extensions.
- To understand the theoretical basics of different computational algorithms for solving linear programming and related problems.

Learning Outcomes:

Students completing this course will be able to:
- Describe the basic concepts of convex analysis and explain the theoretical foundations of various issues related to linear programming modelling.
- Formulate real-world problems as a linear programming model and describe the theoretical workings of the graphical and simplex methods.
- To solve linear programming problems using advanced simplex methods.
- Explain the relationship between a linear program and its dual, including weak and strong duality, complementary slackness, and sensitivity analysis.
- Formulate specialized linear programming problems, namely transportation, transshipment, and assignment problems, and describe the theoretical workings of the solution methods.

Syllabus of DSC-2:

Unit I: Foundations of Linear Programming **(10 hours)**
Theoretical Foundations: Linear independence and dependence of vectors, Basis, Convex sets, Extreme points, Hyperplanes and half spaces, Polyhedral sets and cones. Results based on the above concepts. Introduction to Linear Programming. Problem formulations from different industries.

Unit II: **Simplex Method and Duality** **(13 hours)**
Theory of simplex method. Simplex method, including The Big M and Two-Phase methods. Degeneracy in Simplex Method. Definition and importance of duality. Formulating dual problems. Economic interpretation of duality. Weak and strong duality results. Complementary slackness.

Unit III: **Advanced Simplex Methods, Sensitivity Analysis and Transportation Problem (12 hours)**

Advanced Simplex Methods: Revised simplex method, Dual-simplex method. Sensitivity analysis for Structural and Parameter changes. Transportation problem: Mathematical model, Balanced and unbalanced problems. Degeneracy. Optimality conditions. Methods to find initial and optimal solutions.

Unit IV: **Transshipment and Assignment Problems (10 hours)**

Transshipment Problem: Extension of transportation problem to transshipment model, Formulation and solution. Assignment problem: Mathematical model, Balanced and unbalanced problems. Optimality conditions. Hungarian method for optimal assignment.

Tutorial component (if any) – Yes (15 hours)

Practical component (if any) - Nil

Essential Readings:

1. Bazaraa, M. S., Jarvis, J. J., & Sherali, H. D. (2010). Linear Programming and Network Flows. John Wiley & Sons, Inc.
2. Chandra, S., Jayadeva, & Mehra, A. (2009). Numerical Optimization with Applications. New Delhi: Narosa Publishing House.
3. Hillier, F.S., Lieberman, G. J., & Nag, B. (2021). Introduction to Operations Research, (11th ed.). New Delhi: Tata McGraw Hill (Indian print).
4. Taha, H. A. (2019). Operations Research: An Introduction (10th ed.). New Delhi: Pearson (Indian print).

Suggested Readings: Nil

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

DISCIPLINE SPECIFIC CORE
DSC-3: STATISTICS

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Statistics (DSC-3)	4	3	1	0	-	Graduation

Learning Objectives:

The aim of this course is to:
- Acquaint the students with the fundamental concepts of probability and statistics.
- Provide an understanding of the processes by which real-life statistical problems are analysed.
- Develop an understanding of the role of statistics in Operational Research.

Learning Outcomes:

Students completing this course will be able to:
- Quantify uncertainty using probability, learn how to find probability using the concepts of random variables and distribution functions, obtain characteristics of the underlying distributions, and study functional relationships between two random variables.
- Know various discrete and continuous probability distributions along with their characteristics and identify the situations where they provide realistic models.
- Learn about sampling and sampling distributions along with their characteristics which will help them analyze the population or phenomenon from which the sample is drawn.
- Learn inferential methods wherein the distributional form of population or phenomenon from which the sample is drawn is either known (parametric) or unknown (nonparametric).
- Learn how prior information about a parameter can be used to obtain updated information through Bayesian statistics.

Syllabus of DSC-3:

Unit I: Probability **(9 hours)**
Probability axioms, Conditional probability, Independent events; Random variable, Joint, Marginal and Conditional distributions, Independent random variables, Transformation of one and two- dimensional random variables; Moments: Mean, Variance, Expected value of a function of random variable, Conditional expectation and applications, Probabilistic inequalities, moment inequalities, characteristic functions; Karl Pearson's correlation coefficient and its properties.

Unit II: Distributions and Large Sample Theory **(8 hours)**
Discrete distributions: Degenerate distribution, Bernoulli and Binomial distributions, Hypergeometric distribution, Poisson distribution, Geometric and Negative Binomial distributions; Continuous distributions: Uniform distribution, Normal distribution, Exponential and Gamma distribution, Beta distribution; Weak law of large numbers; Central limit theorems (CLTs).

Unit III: Sampling and Sampling Distributions **(12 hours)**
Population and sample, Statistic, Sample mean, Sample variance, and Sample moments; Order Statistic: Distribution of smallest order statistic and largest order statistic; Chi-Square distribution; F-distribution; Students t- distribution.

Unit IV: Inferential Problems **(16 hours)**
Parametric inference: Problem of point estimation, Method of maximum likelihood estimation, Simple and composite hypotheses, Likelihood ratio tests, Construction of confidence intervals, p value; Parametric tests: normal tests for proportion and mean based on single sample; Chi-Square test for variability; t-test for single mean; t-test for difference of means; paired t-test; F test for equality of variances; Nonparametric tests: run test for randomness, Chi-square test for goodness of fit, one-sample sign test, Wilcoxon signed-rank test; Basics of Bayesian statistics: Bayes theorem, Bayes theorem for future events, Bayes theorem given the data, Conjugate prior distribution: normal conjugate, prior and posterior odds, Bayes factor for simple v/s simple hypothesis, Predictive inference: standard predictive distribution, Laplace's rule of succession.

Tutorial component (if any) - Yes **(15 hours)**

Practical component (if any) - Nil

Essential Readings:

1. Bansal, A. K. (2007). Bayesian Parametric Inference, Narosa Publishing House, New Delhi.
2. Berger, J. 1985. *Statistical Decision Theory and Bayesian Analysis*. New York: Springer-Verlag.
3. Feller, W. (2008). *An introduction to probability theory and its applications (volume 1)* (3rd ed.). Wiley.
4. Fruend, J. E. (2013). *Mathematical statistics with applications* (8th ed.). Pearson Education India.
5. Ross, S. M. (2014). *Introduction to probability models* (11^{th} ed.). Academic press.
6. Levin, R. I., Masood, H. S., Rubin, S. D., & Rastogi, S. (2017). *Statistics for management* (8th ed.). Pearson Education.
7. Mood, A. M., Grabill, F. A., & Boes, D. C. (1974). *Introduction to the theory of statistics* (3rd ed.). McGraw Hill.

Suggested Readings:

1. Blake, I. F. (1987). *An Introduction to Applied Probability*. United States: R.E. Krieger Publishing Company.
2. Black, K. (2013). *Applied business statistics: making better, business decisions* (7^{th} ed.). John Wiley & Sons.
3. Dudewicz, E. J., & Misra S. N. (1988). *Modern mathematical statistics*. Wiley.
4. Goon, A. M., Gupta, A. K., & Dasgupta, B. (1989). *An outline of statistical theory (volume 1)* (2nd ed.). World Press Pvt. Ltd.
5. James, G., Witten D., Hastie, T., Tibshirani, R., Taylor, J. (2021), *An Introduction to Statistical Learning, with Applications in R* (latest edition)., Springer.
6. Rohatgi, V. K., & Ehsanes Saleh, A. K. Md. (2000). *An introduction to probability and statistics* (2nd ed.). Wiley.

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

Discipline Specific Elective – Semester I

DSE - 1(a): Mathematics for Operational Research

DSE - 1(b): Decision Theory

DSE - 1(c): Design Thinking and Innovation

DSE - 1(d): Game Theory with Behavioral Aspects

DSE - 1(e): Simulation Modeling

DSE - 1(f): Software Engineering

DISCIPLINE SPECIFIC ELECTIVE
DSE-1(a): MATHEMATICS FOR OPERATIONAL RESEARCH

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Mathematics for Operational Research (DSE-1(a))	**4**	**3**	**1**	**0**	**-**	Graduation

Learning Objectives:

The learning objectives of this course are as follows:

- To equip students with analytical skills and computational techniques focusing on linear algebra, ordinary differential equations, numerical methods, and graph theory.
- To equip students for analyzing and solving real-world Operational Research problems using these mathematical methods.

Learning Outcomes:

By the end of this course, students will be able to:

- Apply mathematical techniques to solve linear systems and analyze key properties of matrices and vectors.
- Solve and analyze different types of differential equations and apply methods to model real- world phenomena.
- Utilize numerical methods for approximating solutions to complex problems in mathematics and applied fields.
- Understand and apply concepts from graph theory to solve optimization and network-related problems.

Syllabus of DSE-1(a):

Unit I: Linear Algebra **(10 hours)**
Vector Space, Basis and dimension of a vector space, Linear independence and linear dependence of vectors, Solution of a system of linear equations: Gauss Elimination Method (with and without Pivoting), Eigenvalues, Eigenvectors, Eigenspace, Cayley-Hamilton theorem. Diagonalization of matrices. Computation Tools: MATLAB or Python.

Unit II: Ordinary Differential Equations **(12 hours)**
First-order exact differential equations and integrating factors, Linear equations and Bernoulli equations. Higher-order linear differential equations, Homogenous linear equations with constant coefficients, Non-homogenous linear equations, method of undetermined coefficients, and method of variation of parameters. Applications to the Growth and Decay modelling. Computation Tools: MATLAB or Python.

Unit III: Numerical Methods **(12 hours)**
Errors, Roots of Transcendental and Polynomial Equations: Bisection, Newton-Raphson and Secant methods. Order and Rate of convergence. Interpolation: Lagrange and Newton interpolation. Numerical integration: Trapezoidal and Simpson's rules. Computation Tools: MATLAB or Python.

Unit IV: Fundamentals of Graph Theory **(11 hours)**
Graphs: Definition and basic properties, Subgraphs, Complete graphs, and Bipartite graphs. Euler and Hamiltonian paths. Adjacency matrices. Introduction to trees and minimum spanning trees. Applications of graph theory in Network scheduling and optimization. Computation Tools: MATLAB or Python.

Tutorial component (if any) – Yes **(15 hours)**

Practical component (if any) - Nil

Essential Readings:

1. Kolman, Bernard, & Hill, David R. (2001). Introductory Linear Algebra with Applications (7th ed.). Pearson Education, Delhi. First Indian Reprint 2003.
2. Ross, Shepley. L. (2012). Differential Equations (3rd ed.). John Wiley & Sons.
3. Jain, M. K., Iyengar, S. R. K., & Jain, R. K. (2012). Numerical Methods for Scientific and Engineering Computation (6th ed.). New Age International Publisher, India.
4. Gerald, C. F., & Wheatley, P. O. (2008). Applied Numerical Analysis (7th ed.). Pearson Education. India.
5. Rosen, Kenneth H. (2012). Discrete Mathematics and its Applications, with Combinatorics and Graph Theory. (7th ed.). McGraw-Hill Education. Indian Reprint.

Suggested Readings: Nil

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

DISCIPLINE SPECIFIC ELECTIVE
DSE-1(b): DECISION THEORY

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Decision Theory (DSE - 1(b))	**4**	**3**	**1**	**0**	**-**	Graduation

Learning Objectives:

- To teach how optimal choice can be made amongst alternative courses of actions with uncertain consequences using non-probabilistic, probabilistic and utility theory approaches.
- To construct and analyze decision support systems using decision trees and Bayesian decision networks.

Learning Outcomes:

Students completing this course will be able to:
- Analyze problems related to decision making under various environments: under strict uncertainty, under risk: probabilistic approach - without and with data, using utility theory approach.
- Analyze problems related to decision-making involving group of individuals.
- Make decisions with first, second and third degree stochastic dominance decision rules that enable partial ordering amongst competing alternatives; which are helpful in investment decision-making, agriculture, medicine, etc.
- Analyze problems related to sequential decision making under uncertainty.
- Use graphical approach- decision trees and Bayesian decision network for making decisions and apply simulation to decision problems.

Syllabus of DSE-1(b):

Unit I: Introduction (5 hours)
Prescriptive decision analysis; history of decision analysis; Basic elements of decision analysis; Modeling of Decision Problems; Decision Making under Strict Uncertainty: Maximin- Minimax criterion, Maximax-Minimin criterion, Hurwicz criterion, Savage's minimax regret criterion, Laplace equi-likelihood criterion, Some reasonable Properties of a Decision Rule, Analysis of criteria; Strict Uncertainty Impossibility Theorem.

Unit II: Decision Analysis under Risk (15 hours)
Probabilistic Approach- Bayesian Decision Theory: Prior, Posterior, and Preposterior analyses; Decision Analysis without Sampling; Decision Analysis with Sampling; Utility Theory Approach - St. Petersburg Paradox, Expected Utility Principle, Construction of Utility Functions, Risk Attitudes, Properties of Risk Aversion for Monetary Consequences: Utility theory and insurance; Multidimensional utility; Decisions involving groups of individuals; Stochastic Dominance Decision Rules**.**

Unit III: Sequential Decision Making under Uncertainty (10 hours)
Markov Decision Processes (MDPs) - An introduction, Bandit Problems, Decision-Theoretic Bandit process, Finite Horizon MDPs.

Unit IV: Decision Diagrams and Simulation (15 hours)
Decision Trees; Applying simulation to Decision Problems; Bayesian Decision Networks.

Tutorial component (if any) - Yes (15 hours)

Practical component (if any) – Nil

Essential Readings:

1. Fenton, N., & Neil, M. (2013). Risk assessment and decision analysis with bayesian networks. New York: CRC Press, Taylor and Francis Group.
2. Goodwin, Paul and Wright, George (2004). Decision Analysis for Management Judgement, 3rd edition, John Wiley & Sons Ltd.
3. Jensen, F. V., & Nielson, T. D. (2007). Bayesian networks and decision graphs (2nd ed.). New York: Springer Science.
4. Jones, J. M (1977). Introduction to decision theory, irwin series in quantitative analysis for business (1st ed.). New York: Irwin (Richard D.) Inc.
5. Levy, H. (2006). Stochastic dominance – investment decision making under uncertainty (2nd ed.). New York: Springer Science.
6. Parmigiani, G., & Inoue, L. (2009). Decision theory-principles and approaches. UK: John Wiley & Sons Ltd.
7. Sheskin, T. J. (2010). Markov chains and decision processes for engineers and managers. New York: CRC Press, Taylor and Francis Group.

Suggested Readings:

1. Smith, J. Q. (2010). Bayesian decision analysis-principles and practice. UK: Cambridge University Press.
2. Kaas, R., Goovarts, M., Dhaene, J., & Denuit, M. (2001). Modern actuarial risk theory. Netherlands: Kluwer Academic Publishers.

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

DISCIPLINE SPECIFIC ELECTIVE
DSE-1(c): DESIGN THINKING AND INNOVATION

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Design Thinking and Innovation (DSE-1(c))	**4**	**3**	**1**	**0**	**-**	Graduation

Learning Objectives:

The learning objectives of this course are as follows:
- To equip students with knowledge of design thinking process and its application in innovation.
- To equip students with the knowledge of different design thinking approaches and innovation management
- To impart knowledge of data analytics tools for innovation development, positioning and management.

Learning Outcomes:

Students completing this course will be able to:
- Understand the purpose of design thinking and its applications in innovation.
- Learn Design Thinking processes and approaches for ideation, concept development, evaluation, user feedback and iterative design thinking.
- Learn the role of data analytics in innovation development, positioning and management.

Syllabus of DSE 1(c):

Unit I: Introduction to Design Thinking and Innovation (11 hours)
Origin and Purpose of Design and Innovation, Fundamentals of Design Thinking, Design Thinking Process, Key Features of the Design Thinking Process, Cognitive Models for Design Thinking, Innovation and its Importance, Types of Innovation, Design Thinking for Innovation in Physical Products, in Digital Products, in Services, Products and its Levels, Finding Product Market Fit (PMF), Product Life Cycle.

Unit II: Design Thinking Approaches (12 hours)
Design Thinking Approach for Idea Generation: Problem Framing, Idea Generation, Creative & Critical Thinking, Brainstorming-Introduction, Examples, Principles and Process, Methods; Reverse Brainstorming. Design Thinking Approach for Concept Development: Innovation Idea Funnel, Concept Development- Introduction, Process, Tools; Product Concept ideation tools, Storyboard, Sketches and Wireframes, System Map. Design Thinking Approach for Concept Evaluation: Assumption Testing, Kano Model, Value / Ease Matrix, Rapid Prototyping. Design Thinking Approach for Obtaining User Feedback: Minimum Viable Product, Customer Co- Creation. Lifecycle Assessment and Environmental Impact Analysis.

Unit III: Design Thinking Approach for Innovation Development (11 hours)
Innovation, Sources of Innovative Ideas, Jeanne Liedtka's Framework for Design Thinking, Understanding Customers' Expectations, Maslow's Hierarchy of Needs, Types of Data Required to Gather Information Regarding Customers, Research Methods to Gather Customer Data, Value Chain Analysis, Mind Mapping, Value Proposition, Agile Methodologies in Design and Development.

Unit IV: Innovation Management (11 hours)
Data Analytics for Innovation: Customer Segmentation and Targeting, Data-driven Design Decision Making, Positioning and Differentiation Go-to-market Strategy, Adoption of Innovation, Innovation Adopters, Models of innovation adoption & management.

Tutorial component (if any) – Yes (15 hours)

Practical component (if any) – Nil

Essential Readings:

1. Christian Müller-Roterberg, (2018). Handbook of Design Thinking, Kindle Direct Publishing.
2. Gavin Ambrose, Paul Harris (2010). Basics Design: Design Thinking, AVA Publishing.
3. Tim Brown (2009). Change by design, 1st Ed., Harper Bollins.
4. Kelly Tom (2001). The Art of Innovation, Currency.
5. Christensen, Clayton M. (1997). The Innovator's Dilemma: When New Technologies Cause Great Firms to Fail. Boston, MA: Harvard Business School Press.
6. Lilien, Gary.L, Kotler P.& Moorthy K. Sridhar (1998). Marketing Models, Prentice Hall India Learning Private Limited.

Suggested Readings:

1. Idris Mootee (2013). Design Thinking for Strategic Innovation: What They Can't Teach You at Business or Design School", John Wiley & Sons.
2. Maurício Vianna, Ysmar Vianna, Isabel K. Adler, Brenda Lucena, Beatriz Russo (2011). Design thinking: Business Innovation MJV Press.

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

DISCIPLINE SPECIFIC ELECTIVE
DSE-1(d): GAME THEORY WITH BEHAVIORAL ASPECTS

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Game Theory with Behavioral Aspects (DSE-1(d))	**4**	**3**	**1**	**0**	**-**	Graduation

Learning Objectives:

The learning objectives of this course are as follows:

- To provide an in-depth exploration of game theory and its application to business and economics, including 2-persons and n-persons.
- To understand behavioral game theory, highlighting how psychological factors affect strategic decisions.

Learning Outcomes:

Students completing this course will be able to:

- Understand the basic concepts of game theory and its relevance to business and economics.
- Study famous games such as Prisoner's Dilemma, The Ultimatum Game, The Tragedy of the Commons, The Battle of the Sexes and The Stag Hunt.
- Analyse and develop core strategies for various types of games, including 2-persons and n- persons.
- Apply game-theoretic concepts to real-world problems in business, economics, and society.
- Use behavioral scenarios in strategic decision-making.

Syllabus of DSE-1(d):

Unit I: Foundations of Game Theory **(10 hours)**
Introduction to Game Theory: Overview of game theory, Importance and applications in business. Key concepts: players, strategies, payoffs, and value of the game. Non-cooperative Games: Examples of discrete and continuous static games involving two-persons and n- persons. Famous games: Prisoner's Dilemma, The Ultimatum Game, and The Tragedy of the Commons.

Unit II: **Zero-Sum Matrix Games** **(11 hours)**
Two-person zero-sum matrix game: Minimax and Maximin rules, convex and concave games, Dominant and dominated strategies. Solution of matrix games of size **2 x 2, 2 x n, m x 2, and m x n**.

Unit III: Nash Equilibrium and Game Theory Applications **(12 hours)**
Nash Equilibrium: Definition, properties, and significance. Computation of equilibria using optimality conditions. Solution of Bi-matrix and Symmetric matrix games. Game theory applications in advertising, strategic positioning, and oligopoly models, voting systems, bidding, and negotiation.

Unit IV: Cooperative Games and Behavioral Insights (12 hours)
Introduction to Cooperative Games. Solutions based on characteristic functions: The core and the Shapley values. Behavioral Game Theory: Bounded rationality, psychological aspects of decision-making, biases, fairness, reciprocity, and social preferences.

Tutorial component (if any) – Yes (15 hours)

Practical component (if any) - Nil

Essential Readings:

1. Mastumoto, A., & Szidarovszky, F. (2016). Game Theory and Its Applications. Springer.
2. Mazalov, V. (2014). Mathematical Game Theory and Applications. John Wiley & Sons, Inc.
3. Thie, P. R. & Keough, G. E. (2008). An Introduction to Linear Programming and Game Theory*,* John Wiley & Sons, Inc.

Suggested Readings: Nil

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

DISCIPLINE SPECIFIC ELECTIVE
DSE-1(e): SIMULATION MODELING

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Simulation Modeling (DSE-1(e))	**4**	**3**	**1**	**0**	**-**	Graduation

Learning Objectives:

The learning objectives of this course are as follows:
- To provide detailed understanding of simulation concepts and its applications.
- To help students to design and analyse simulation models with its application on real world problems.

Learning Outcomes:

Students completing this course will be able to:
- Understand the fundamentals of simulation and model building.
- Apply different methods of random number generation to generate discrete and continuous random variable
- Conduct analysis of simulation models.
- Understand and apply principles of Monte Carlo simulation.
- Decode case studies to discover various simulation applications.

Syllabus of DSE-1(e):

Unit I: Fundamentals of Simulation and Modeling (12 hours)
Introduction to Simulation: concept of model and model building, Concept and terminologies of simulation, Process of simulation, Advantages and limitations of simulation, Classification of simulation models: physical and mathematical models, static and dynamic model, discrete and continuous models, deterministic and stochastic models. Application areas of simulation.

Unit II: Random Number Generation (11 hours)
Properties of random numbers, Generation of pseudo random numbers, Techniques of generating discrete and continuous random variables: inverse transformation, direct transformation, rejection method, hazard rate method. Test for random numbers.

Unit III: Design and Analysis of Simulation Models (12 hours)
Data collection, identifying distributions with data, Parameter estimation, Goodness of fit tests, selecting input models without data, Steady-state simulation, Terminating simulation, Confidence interval estimation, Output analysis for steady state simulation. Simulation run statistics, Replication of runs, Elimination of initial bias.

Unit IV: Monte Carlo Methods, Simulation Tools, and Applications (10 hours)
Monte Carlo simulation. Simulation tools: General-purpose simulation tools, Case studies of different types of simulation.

Tutorial component (if any) - Yes (15 hours)

Practical component (if any) - Nil

Essential Readings:

1. Ross, S., *Simulation*, Academic Press, 5th Edition 2012.
2. A.M. Law and W.D. Kelton, *Simulation and Modeling and analysis*, 5th Edition, 2015.
3. Evans, J. R., & Olson, D. L. *Introduction to simulation and risk analysis*. Prentice-Hall, Inc. 2nd Edition, 2001.
4. Geoffrey Gordon: *System Simulation*, 2nd Edition, 2002.
5. Frank L. Severance, *System Modeling and Simulation: An Introduction*, Wiley, 1st Edition, 2001.
6. J Banks, J. S. Carson II, B. L. Nelson, D. M. Nicol, 2010, Discrete Events System Simulation, 5th Edition, Prentice Hall.

Suggested Readings: Nil

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

DISCIPLINE SPECIFIC ELECTIVE
DSE-1(f): SOFTWARE ENGINEERING

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Software Engineering (DSE-1(f))	**4**	**3**	**1**	**0**	**-**	Graduation

Learning Objectives:

The learning objectives of this course are as follows:
- To teach about the software development life cycle and related stages.
- To teach the concepts of software requirements analysis followed by design preparation.
- To teach various testing pedagogies available for software verification and validation.
- To teach how to quantitatively assess a software system's reliability by using different software reliability growth models.
- To teach about the software maintenance phenomenon.

Learning Outcomes:

Students completing this course will be able to:
- Understand the software development life cycle, its various stages, and different approaches for software development, such as the waterfall and evolutionary models.
- Know about recent advances in the software development process.
- Perform software requirements analysis, system design preparation.
- Understand Software Project management activities, including planning, scheduling, risk management, etc.
- Understand software testing approaches testing and integration testing, alpha and beta testing, System testing, Functional testing, Structural testing.
- Understand quality control and how to ensure good quality software.
- Develop and validate the mathematical models for in software reliability assessment and prediction.
- Understand the concept of multi-up-gradation for software maintenance.

Syllabus of DSE-1(f):

Unit I: Introduction to Software Engineering **(8 hours)**
Introduction to Software Engineering and related Principles, Software metrics and measurement, monitoring and control, Software development life-cycle Models: Software development life-cycle, Waterfall model, prototyping model, Incremental model, Iterative enhancement Model, Spiral model, Open Source Software and its life cycle.

Unit II: Stages of Software Development Process (20 hours)
Software Requirements: Analysis, Specification & Elicitation Techniques, requirements validation. System Design: Design Principles, Problem partitioning, abstraction, design specification, Cohesiveness and Coupling Software Project Management: Project planning, Software Metrics, Cost estimation using constructive cost models (Basic, Intermediate, and Detailed COCOMO), Risk management activities Software Testing**:** Verification and validation, code inspection, test plan, test case specification. Levels of testing: Unit, Integration Testing, Top-down and bottom-up integration testing, Alpha and Beta testing, System testing and debugging. Functional testing, Structural testing, Software testing strategies.

Unit III: Software Reliability (10 hours)
Modeling Software Reliability and its uses, Difference between hardware and software Reliability, Non-homogeneous Poisson Process models, Imperfect Debugging models, testing effort-based modeling, the concept of change point, Release Time problems based on Cost Criterion, Reliability Criterion, Cost and Reliability Criteria, Reliability of modular software, Resource Allocation Problem

Unit IV: Software Quality Assurance & Maintenance (7 hours)
Software quality, ISO 9000 certification for the software industry, SEI capability maturity model, Software implementation and integration, Software Maintenance models through multi- up-gradation concept.

Tutorial component (if any) - Yes (15 hours)

Practical component (if any) - Nil

Essential Readings:

1. Aggarwal, K. K., & Singh Y. (2005). Software engineering, New Age International.
2. Kapur, P., Pham, H., Gupta, A., & Jha, P. C. (2011). Software reliability assessment with OR applications. London: Springer-Verlag.
3. Pressman, R. S. (2005). Software engineering: a practitioner's approach. Palgrave Macmillan.
4. Wang, H., & Pham, H. (2010). Reliability and optimal maintenance. London: Springer- Verlag.
5. Yamada, S. (2014). Software reliability modeling: fundamentals and applications. Tokyo: Springer.

Suggested Readings: Nil

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

Skill Based Course – Semester I

SBC - 1: Database Management System

SKILL ENHANCEMENT COURSE
SEC-1: DATABASE MANAGEMENT SYSTEM

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Database Management System (SBC-1)	**2**	**1**	**0**	**1**	**-**	Graduation

Learning Objectives:

- The objective of the course is to present an introduction to Database Management Systems (DBMS), emphasizing how to organize, maintain, and retrieve information efficiently and effectively from a DBMS.
- The course shall provide a technical overview of database management systems, enabling the students to understand the logical design of the database using data modeling concepts and thereby manipulating any database using SQL.

Learning Outcomes:

Students completing this course will be able to:
- Describe the fundamental elements of relational database management systems.
- Utilize a wide range of features available in a DBMS package.
- Analyze database requirements and determine the entities involved in the system and their relationship to one another.
- Develop the logical design of the database using data modeling concepts.
- Manipulate a database using SQL.

Syllabus of SEC-1:

Unit I: Introduction to Databases and basics of SQL (5 Theory hours + 10 Practical hours)
Database Management System- Introduction, concepts and architecture, Characteristics, Advantages of Using DBMS, Brief History of Database Application, Data Models, Schemas, Instances, Three-Schema Architecture and Data Independence, Database Languages and Interfaces, Classification of Database Management Systems. DDL commands in SQL-Table creation, alteration, defining constraints – Primary key, foreign key, unique, not null, check,

Unit II: Relational Algebra, Data Modeling and Database Design
(5 Theory hours + 5 Practical hours)
Structure of Relational databases, Domains, Relations, Relational Algebra and related queries, relational calculus, Entity Types, Entity Sets, Attributes, and Keys; Relationships and Participation; Codd's Rules; ER Diagrams and Schema; EER Model; Relational Model Concepts; Functional Dependencies; Normalization (1NF, 2NF, 3NF, BCNF).

Unit III: Structured Query Language **(2 Theory hours + 15 Practical hours)**
SQL Commands: DML-update, delete, select – all columns, specific columns, unique records, conditional select, in clause, between clause, limit, aggregate functions (count, min, max, avg,sum), DQL, DCL, and TCL, group by clause, having clause. Use of group by, having, order by, join and its types, String Functions, Math Functions and Date Functions in SQL, set operations, sub queries, correlated sub-queries.

Unit IV: Transaction Management and Query Optimization **(3 Theory hours)**
Transaction Concepts; Transaction States; Properties of a Transaction; Concurrency Control; Disk Storage; RAID; Query Processing and Optimization; Distributed Databases.

Practical component (if any):

Students shall indulge in performing Practical in Computer Lab on SQL according to the above theory syllabus.

Essential Readings:

1. Dasai, B.C. (1998). Database System, BPB.
2. Date, C. J. (2006). An introduction to database systems. Pearson Education India.
3. Elmasri, R., & Navathe, S. (2011). Fundamentals of database systems (6th ed.), Addison- Wesley Publishing Company.
4. Ramakrishnan, R., & Gehrke, J. (2000). Database management systems. McGraw Hill. Schneider, D. I. (2013). An introduction to programming using visual basic 2012. Prentice Hall Press.
5. Silberschatz, A., Korth, H. F., & Sudarshan, S. (1997). Database system concepts (Vol. 4). New York: McGraw-Hill.

Suggested Readings: Nil

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

Generic Elective - Semester I

GE - 1(a): Inventory Management

GE - 1(b): Linear Programming & Extensions

GE - 1(c): Game Theory with Behavioral Aspects

GE - 1(d): Simulation Modeling

GENERIC ELECTIVE
GE-1(a): INVENTORY MANAGEMENT

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Prerequisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Inventory Management (GE-1(a))	4	3	1	0	-	Graduation

Learning Objectives:

The learning objectives of this course are as follows:

- To impart knowledge of the role of inventory in organizations and core concepts of inventory management with an understanding of fundamental inventory control procedures and their usage.
- To provide students with a thorough understanding of basic inventory models and a rigorous mathematical framework to develop mathematical models to analyze and optimize inventory systems.
- To provide the students with a comprehensive study of various application areas of inventory models through case studies and relevant examples.

Learning Outcomes:

Students completing this course will be able to:

- Identify the goals and objectives of inventory management and describe the importance of stocks in an organization and the reasons for holding stock.
- Explain the various costs related to the inventory system.
- Understand the various selective inventory control techniques and their applications.
- Capability to develop deterministic inventory models: economic order quantity and its extensions, All units and incremental quantity discounts models, Joint and Individual order policies, and Production scheduling models.
- Understand and develop stochastic inventory models and setting safety stocks. Apply and extend inventory models to analyse real-world systems.

Syllabus of GE-1(a):

Unit I: Introduction to Inventory Systems (9 hours)

Analytical structure of Production and Inventory problems. Objectives of Inventory Management. Factors influencing inventories. Inventory-related costs. Properties of Inventory systems. Selective Inventory control techniques and their applications. Concept of Lead Time. Introduction to Just in Time (JIT) and Vendor Managed Inventory (VMI).

Unit II: Deterministic Inventory Models (15 hours)

Deterministic inventory models, economic order quantity and its extensions: without and with lead time. Finite replenishment rate Inventory models without and with planned shortages. Inventory models with partial backlogging and lost sales. Discrete Demand Model. Multi-item Inventory models with constraints. Quantity discounts: All units and incremental. Joint and Individual Ordering Policies.

Unit III: Production Planning Models **(9 hours)**
Aggregate Production Planning Models: Fixed workforce model. Variable workforce model. Dynamic lot size models: Wagner-Whitin Algorithm, Silver-Meal heuristic.

Unit IV: Stochastic Inventory Models **(12 hours)**
Stochastic Inventory models, Newsvendor model and its extensions: Instantaneous and uniform demand with discrete and continuous cases; without and with lead time. Transformations for the equivalence of instantaneous and uniform demand models. Power demand pattern inventory model. Periodic review models. Safety stocks, Service levels and reorder level.

Tutorial component (if any) - Yes **(15 hours)**

Practical component (if any) - Nil

Essential Readings:

1. Axsäter, S. (2015). *Inventory Control.* (Third Edition). Springer.
2. Hadley, G., & Whitin, T. M. (1963). *Analysis of Inventory Systems*. Prentice-Hall.
3. Johnson, L.A., & Montgomery, D.C. (1974) *Operations Research in Production Planning, Scheduling and Inventory Control*. Wiley, New York.
4. Muckstadt, J. A., & Sapra, A. (2010). *Principles of Inventory Management: When You Are Down to Four, Order More*. Springer Science & Business Media.
5. Naddor, E. (1966). *Inventory Systems*. Wiley.
6. Silver, E. A., Pyke, D. F., & Peterson, R. (1998). *Inventory Management and Production Planning and Scheduling*. (Third Edition). Wiley.
7. Waters, D. (2008). *Inventory Control and Management.* (Second Edition). John Wiley & Sons.
8. Zipkin, H. P. (2000). *Foundations of Inventory Management*. McGraw-Hill.

Suggested Readings: Nil

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

GENERIC ELECTIVE
GE-1(b): LINEAR PROGRAMMING AND EXTENSIONS

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Linear Programming and Extensions (GE-1(b))	**4**	**3**	**1**	**0**	**-**	**Graduation**

Learning Objectives:

The learning objectives of this course are as follows:

- To impart knowledge of formulating practical problems using the linear programming method and its extensions.
- To understand the theoretical basics of different computational algorithms for solving linear programming and related problems.

Learning Outcomes:

Students completing this course will be able to:

- Describe the basic concepts of convex analysis and explain the theoretical foundations of various issues related to linear programming modelling.
- Formulate real-world problems as a linear programming model and describe the theoretical workings of the graphical and simplex methods.
- To solve linear programming problems using advanced simplex methods.
- Explain the relationship between a linear program and its dual, including weak and strong duality, complementary slackness, and sensitivity analysis.
- Formulate specialized linear programming problems, namely transportation, transshipment, and assignment problems, and describe the theoretical workings of the solution methods.

Syllabus of GE-1(b):

Unit I: Foundations of Linear Programming (10 hours)
Theoretical Foundations: Linear independence and dependence of vectors, Basis, Convex sets, Extreme points, Hyperplanes and half spaces, Polyhedral sets and cones. Results based on the above concepts. Introduction to Linear Programming. Problem formulations from different industries.

Unit II: **Simplex Method and Duality (13 hours)**
Theory of simplex method. Simplex method, including The Big M and Two-Phase methods. Degeneracy in Simplex Method. Definition and importance of duality. Formulating dual problems. Economic interpretation of duality. Weak and strong duality results. Complementary slackness.

Unit III: **Advanced Simplex Methods, Sensitivity Analysis and Transportation Problem (12 hours)**

Advanced Simplex Methods: Revised simplex method, Dual-simplex method. Sensitivity analysis for Structural and Parameter changes. Transportation problem: Mathematical model, Balanced and unbalanced problems. Degeneracy. Optimality conditions. Methods to find initial and optimal solutions.

Unit IV: **Transshipment and Assignment Problems (10 hours)**

Transshipment Problem: Extension of transportation problem to transshipment model, Formulation and solution. Assignment problem: Mathematical model, Balanced and unbalanced problems. Optimality conditions. Hungarian method for optimal assignment.

Tutorial component (if any) – Yes (15 hours)

Practical component (if any) - Nil

Essential Readings:

1. Bazaraa, M. S., Jarvis, J. J., & Sherali, H. D. (2010). Linear Programming and Network Flows. John Wiley & Sons, Inc.
2. Chandra, S., Jayadeva, & Mehra, A. (2009). Numerical Optimization with Applications. New Delhi: Narosa Publishing House.
3. Hillier, F.S., Lieberman, G. J., & Nag, B. (2021). Introduction to Operations Research, (11th ed.). New Delhi: Tata McGraw Hill (Indian print).
4. Taha, H. A. (2019). Operations Research: An Introduction (10th ed.). New Delhi: Pearson (Indian print).

Suggested Readings: Nil

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

GENERIC ELECTIVE
GE-1(c): GAME THEORY WITH BEHAVIORAL ASPECTS

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Game Theory with Behavioral Aspects (GE-1(c))	**4**	**3**	**1**	**0**	**-**	Graduation

Learning Objectives:

The learning objectives of this course are as follows:
- To provide an in-depth exploration of game theory and its application to business and economics, including 2-persons and n-persons.
- To understand behavioral game theory, highlighting how psychological factors affect strategic decisions.

Learning Outcomes:

Students completing this course will be able to:
- Understand the basic concepts of game theory and its relevance to business and economics.
- Study famous games such as Prisoner's Dilemma, The Ultimatum Game, The Tragedy of the Commons, The Battle of the Sexes and The Stag Hunt.
- Analyse and develop core strategies for various types of games, including 2-persons and n- persons.
- Apply game-theoretic concepts to real-world problems in business, economics, and society.
- Use behavioral scenarios in strategic decision-making.

Syllabus of GE-1(c):

Unit I: Foundations of Game Theory **(10 hours)**
Introduction to Game Theory: Overview of game theory, Importance and applications in business. Key concepts: players, strategies, payoffs, and value of the game. Non-cooperative Games: Examples of discrete and continuous static games involving two-persons and n- persons. Famous games: Prisoner's Dilemma, The Ultimatum Game, and The Tragedy of the Commons.

Unit II: **Zero-Sum Matrix Games** **(11 hours)**
Two-person zero-sum matrix game: Minimax and Maximin rules, convex and concave games, Dominant and dominated strategies. Solution of matrix games of size **2 x 2, 2 x n, m x 2, and m x n**.

Unit III: Nash Equilibrium and Game Theory Applications **(12 hours)**
Nash Equilibrium: Definition, properties, and significance. Computation of equilibria using optimality conditions. Solution of Bi-matrix and Symmetric matrix games. Game theory applications in advertising, strategic positioning, and oligopoly models, voting systems, bidding, and negotiation.

Unit IV: Cooperative Games and Behavioral Insights **(12 hours)**
Introduction to Cooperative Games. Solutions based on characteristic functions: The core and the Shapley values. Behavioral Game Theory: Bounded rationality, psychological aspects of decision-making, biases, fairness, reciprocity, and social preferences.

Tutorial component (if any) – Yes **(15 hours)**

Practical component (if any) - Nil

Essential Readings:

1. Mastumoto, A., & Szidarovszky, F. (2016). Game Theory and Its Applications. Springer.
2. Mazalov, V. (2014). Mathematical Game Theory and Applications. John Wiley & Sons, Inc.
3. Thie, P. R. & Keough, G. E. (2008). An Introduction to Linear Programming and Game Theory, John Wiley & Sons, Inc.

Suggested Readings: Nil

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

GENERIC ELECTIVE
GE-1(d): SIMULATION MODELING

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Simulation Modeling (GE-1(d))	**4**	**3**	**1**	**0**	**-**	Graduation

Learning Objectives:

The learning objectives of this course are as follows:

- To provide detailed understanding of simulation concepts and its applications.
- To help students to design and analyse simulation models with its application on real world problems.

Learning Outcomes:

Students completing this course will be able to:

- Understand the fundamentals of simulation and model building.
- Apply different methods of random number generation to generate discrete and continuous random variable
- Conduct analysis of simulation models.
- Understand and apply principles of Monte Carlo simulation.
- Decode case studies to discover various simulation applications.

Syllabus of GE-1(d):

Unit I: Fundamentals of Simulation and Modeling (12 hours)
Introduction to Simulation: concept of model and model building, Concept and terminologies of simulation, Process of simulation, Advantages and limitations of simulation, Classification of simulation models: physical and mathematical models, static and dynamic model, discrete and continuous models, deterministic and stochastic models. Application areas of simulation.

Unit II: Random Number Generation (11 hours)
Properties of random numbers, Generation of pseudo random numbers, Techniques of generating discrete and continuous random variables: inverse transformation, direct transformation, rejection method, hazard rate method. Test for random numbers.

Unit III: Design and Analysis of Simulation Models (12 hours)
Data collection, identifying distributions with data, Parameter estimation, Goodness of fit tests, selecting input models without data, Steady-state simulation, Terminating simulation, Confidence interval estimation, Output analysis for steady state simulation. Simulation run statistics, Replication of runs, Elimination of initial bias.

Unit IV: Monte Carlo Methods, Simulation Tools, and Applications (10 hours)
Monte Carlo simulation. Simulation tools: General-purpose simulation tools, Case studies of different types of simulation.

Tutorial component (if any) - Yes (15 hours)

Practical component (if any) - Nil

Essential Readings:

1. Ross, S., *Simulation*, Academic Press, 5th Edition 2012.
2. A.M. Law and W.D. Kelton, *Simulation and Modeling and analysis*, 5th Edition, 2015.
3. Evans, J. R., & Olson, D. L. *Introduction to simulation and risk analysis*. Prentice-Hall, Inc. 2nd Edition, 2001.
4. Geoffrey Gordon: *System Simulation*, 2nd Edition, 2002.
5. Frank L. Severance, *System Modeling and Simulation: An Introduction*, Wiley, 1st Edition, 2001.
6. J Banks, J. S. Carson II, B. L. Nelson, D. M. Nicol, 2010, Discrete Events System Simulation, 5th Edition, Prentice Hall.

Suggested Readings: Nil

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

Discipline Specific Core - Semester II

DSC - 4: Optimization Techniques

DSC - 5: Queueing Theory

DSC - 6: Python Programming for Decision-Making

DISCIPLINE SPECIFIC CORE
DSC-4: OPTIMIZATION TECHNIQUES

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Optimization Techniques (DSC-4)	**4**	**3**	**1**	**0**	**-**	Graduation

Learning Objectives:

The learning objectives of this course are as follows:
- To comprehensively understand optimization techniques and their applications in science, engineering, economics, and management.
- To analyze and apply nonlinear optimization, integer linear, and linear goal programming methods to complex decision-making situations in diverse fields.

Learning Outcomes:

Students completing this course will be able to:
- Explain the concepts of convex functions and their properties and describe the convex optimization problem.
- Describe the optimality conditions for unconstrained and constrained optimization problems.
- Demonstrate the formulations of real-world situations as integer linear programming problems and describe the theoretical workings of the solution methods.
- Demonstrate the formulations of real-world situations as quadratic programming problems and describe the theoretical workings of the solution methods.
- Demonstrate the formulations of real-world situations as linear goal programming problems and describe the theoretical workings of the solution methods.

Syllabus of DSC-4:

Unit I: Unconstrained Optimization and Convex Functions (10 hours)
Unconstrained optimization problems: Local and global extrema, necessary and sufficient optimality conditions, line search method, and gradient descent method. Introduction to Convex Functions. Properties of Convex Functions. Applications in Optimization.

Unit II: **Constrained Optimization: Optimality, Duality and Methods (12 hours)**
Constrained optimization problems: Lagrange multipliers, Karush-Kuhn-Tucker optimality conditions. Wolfe Dual. Weak and strong duality results. Quadratic Programming: Wolfe method, sequential linear programming (Frank-Wolfe) method and reduced gradient method.

Unit III: Integer Linear Programming: Concepts and Methods (12 hours)
Introduction to Integer linear programming. Modelling pure and mixed integer linear programming for real-world problems, including manufacturing, logistics, finance, telecommunication, and scheduling. Solution techniques: Branch and bound. Gomory's cutting plane algorithm. 0-1 programming problem. E-Bala's additive algorithm.

Unit IV: Goal Programming: Concepts and Methods (11 hours)
Introduction to Goal Programming. Types of goal programming: linear goal programming, Archimedean goal programming, and Preemptive goal programming. Solution techniques: Graphical method and Lexicographic Simplex method. Real-world modelling applications using linear goal programming in manufacturing, logistics, finance, telecommunications, and scheduling.

Tutorial component (if any) – Yes (15 hours)

Practical component (if any) - Nil

Essential Readings:

1. Antoniou, A., & Lu, Wu-Sheng (2021). Practical Optimization: Algorithms and Engineering Applications (2nd ed.). Springer.
2. Bazaraa, M. S., Sherali, H. D., & Shetty, C. M. (2006). Nonlinear programming; Theory and Algorithms (3rd ed.). John Wiley & Sons, Inc.
3. Chandra, S., Jayadeva, & Mehra, A. (2009). Numerical Optimization with Applications, Narosa Publishing House.
4. Hillier, F.S., Lieberman, G. J., & Nag, B. (2021). Introduction to Operations Research (11th ed.). Tata McGraw Hill.
5. Taha, H. A. (2019). Operations Research: An Introduction (10th ed.). Pearson.

Suggested Readings: Nil

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

DISCIPLINE SPECIFIC CORE
DSC-5: QUEUEING THEORY

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Queueing Theory (DSC-5)	4	3	1	0	-	Graduation

Learning Objectives:

The learning objectives of this course are as follows:
- To impart knowledge of queueing theory and provide the students with a rigorous framework which would enable them to model and analyze queueing systems.
- To provide the students necessary mathematical support and confidence to the students to tackle real life problems.
- To explore both theory and application of fundamental and advanced models in this field.

Learning Outcomes:

Students completing this course will be able to:
- Gain a deep understanding of the theoretical background of queueing systems.
- Understand and compute quantitative metrics of performance for queueing systems.
- Apply and extend queueing models to analyze real world systems.

Syllabus of DSC-5:

Unit I: Introduction **(10 hours)**
Basic Concepts of Stochastic Process, Markov Chains, Introduction to Queueing Systems, Characteristics of Queueing Systems, Counting Process, Poisson Process (Pure-Birth Process and Pure-Death Process), Expected Measures of System Performance, Queueing Simulation: Data Generation and Book-Keeping.

Unit II: Markovian Queueing Models **(15 hours)**
General Birth-Death Process, Infinite Capacity Queues: Single-Server Queues (M/M/1) and Multi-Server Queues (M/M/c), Finite Capacity Queues: M/M/1/K, M/M/c/K and M/M/c/c (Erlang's Loss System), Queues with Unlimited Service ($M/M/\infty$), Finite-Source Queues, Queues with State-Dependent Service, Queues with Impatience (M/M/1 Balking and M/M/1 Reneging).

Unit III: Advanced Markovian Queueing Models **(15 hours)**
Queues with Bulk Arrival ($M^{[X]}/M/1$), Queues with Bulk Service ($M/M^{[Y]}/1$), Erlangian Queueing Models ($M/E_k/1$ and $E_k/M/1$), Concept of Imbedded Markov Chains, Queues with General Arrivals (G/M/1), Queues with General Service (M/G/1).

Unit IV: Decision Problems in Queueing Systems (5 hours)
Design and Control Problems in Queueing Systems, Queueing System with Server Vacation, Queueing System with Removable Server.

Tutorial component (if any) - Yes (15 hours)

Practical component (if any) - Nil

Essential Readings:

1. Bhat, U. N. (2015). An introduction to Queueing Theory: Modelling and Analysis in Applications (Statistics for Industry and Technology) (2nd Edition). Birkhauser Boston.
2. Cooper, R.B. (1981). Introduction to Queueing Theory (2nd Edition). Elsevier North Holland.
3. Cox, D. R. and Smith, W. L. (1991). Queues. Chapman and Hall/CRC.
4. Gross, Donald, Shortle , John F., Thompson, James M., and Harris, Carl M. (2018). Fundamentals of Queueing Theory (5th Edition), John Wiley and Sons Inc. Pte. Ltd.
5. Kleinrock L. (1975). Queueing Systems, Volume 1: Theory, John Wiley.
6. Medhi, J. (2003). Stochastic Models in Queueing Theory (2nd Edition), Academic Press.
7. Prabhu, N. U. (2012). Foundations of Queueing Theory (International Series in Operations Research & Management Science), Springer (Softcover reprint of the original 1st ed. 1997 edition.)
8. Satty, T. L. (1983). Elements of Queueing Theory with Applications, Dover Publications, NY. (Reprint Edition)

Suggested Readings: Nil

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

DISCIPLINE SPECIFIC CORE
DSC-6: PYTHON PROGRAMMING FOR DECISION-MAKING

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Python Programming for Decision-Making (DSC-6)	**4**	**3**	**0**	**1**	**-**	Graduation

Learning Objectives:

The learning objectives of this course are as follows:
- To introduce the basic concepts of Python programming required for data science.
- To teach the students about Python's ability to handle different data formats such as numbers, strings, lists, dictionaries, sets, tuples, etc.
- The students will be made familiar with the concepts of loops, modularization of code using built-in functions and user-defined functions will also be explained.
- Introduce the basics of various useful libraries to equip the students with modern computing skills.

Learning Outcomes:

Students completing this course will be able to:
- Learn Python installation, configuration and understand scripting using python.
- Understand different data types and arithmetical, logical and relational expressions in Python.
- Understand the control structures and functions in Python and handle simple data structures, lists, dictionaries, sets and tuples and modularize the code using built-in functions and user- defined functions.
- Implement Object Oriented Programming concepts in Python
- Write clean and efficient Python code for data analysis and build pipelines from raw data to insights.
- Visualize and communicate data findings.
- Understand the concept of various algorithms and work on real-world projects.

Syllabus of DSC-6:

Unit I: Introduction to Python Programming (11 hours)
Familiarization with the basics of Python programming, interactive mode, and script mode, Structure of a Program, process of writing a program, script execution, and debugging errors. Identifiers, Keywords, Constants, Variables, and data types. Arithmetic operators, Relational operators, Logical operators, Ternary operators, and Bitwise operators. Input and Output Statements, Control Structures. Introduction to functions: modules, built-in and user-defined functions: impeding modules, invoking built-in functions. User-defined functions: Parameters, scope of variables, passing parameters, void functions, and functions returning values.

Unit II: Data Structures (Strings, Lists, Tuples and Dictionary) (11 hours)
Strings: initializing strings and accessing the elements, string operations, built-in string functions, and methods. Lists: concepts of mutable lists; List operations: creating, initializing, accessing, traversing, appending/inserting, searching, and deleting elements; list functions (inbuilt and user-defined). Tuples: Concepts of immutable, creating, initializing, accessing elements; tuple assignment, slices, and indexing; tuple functions. Dictionary: Concept of key- value pair, creating, initializing, accessing, traversing, appending, updating and deleting elements; dictionary functions and methods.

Unit III: Object Orientation in Python (9 hours)
Creation of instance variables in Python, self-keyword in Python, static and non-static methods in Python, OOPS concept (Encapsulation, polymorphism, Abstraction, Inheritance), concept of Classes, Exception handling.

Unit IV: Advanced Python Concepts (14 hours)
Working with libraries: Pandas, NumPy, Matplotlib, and Sci-kit learn; Pandas: Working with Data Frame, importing from csv files, Numpy: Array and Matrix Operations, Matplotlib: Plotting graphs for various mathematical and statistical functions, Sci-kit learn: Data Pre-processing, Curve Fitting and Regression—Python libraries for Optimization: SciPy and PuLP.

Practical component (if any): Yes (30 hours)

Students shall indulge in performing Practical in Computer Lab on Python Software according to the above theory syllabus.

1. Write a program to enter name and display as "Hello, Name".
2. Write a menu driven program to enter two numbers and print the arithmetic operations like
   a. + b. – c. * d. / e. // f. %.
3. Write a program to compute the roots of a quadratic equation.
4. Write a menu driven program to reverse the entered numbers and print the sum of digits entered.
5. Write a menu driven program to enter the number and print whether the number is odd or even prime.
6. Write a program to find maximum out of entered 3 numbers
7. Write a program to display ASCII code of a character and vice versa.
8. Write a program to check if the entered number is Armstrong or not.
9. Write a program to find factorial of the entered number using recursion.
10. Write a program to enter the number of terms and to print the Fibonacci Series.
11. Write a program to enter the numbers and to print greatest number using loop.
12. Write a program to enter the string and to check if it's palindrome or not using loop.
13. Write a program to enter the 5 subjects' numbers and print the grades A/B/C/D/E.
14. Write a program in python language to display the given pattern:

```
            5
         4  5
      3  4  5
   2  3  4  5
1  2  3  4  5
```

15. Write a python function sin (x, n) to calculate the value of sin(x) using its Taylor series expansion up to n terms.
    $$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots,$$
16. Write a program to determine EOQ using various inventory models.
17. Write a program to determine different characteristics using various queuing models.
18. Write a program to implement Inheritance. Create a class Employee inherit two classes Manager and Clerk from Employee.
19. Write a program to fit Poisson distribution on a given data.
20. Write a program to implement linear regression using python.
21. Write a program to perform read and write operation with .csv file.
22. Write a program to enter multiple values-based data in multiple columns/rows and show that data in Python using DataFrames and pandas.
23. Write a program in python to perform various statistical measures using pandas.
24. Write a program to plot a bar chart in python to display the result of a school for five consecutive years.
25. Write a program in python to plot a graph for the function $y = x^2$
26. Write programs related to creating and modifying List, Tuple and Dictionary.
27. Write programs to find correlation between dependent and independent variables.
28. Write programs for data visualization (Charts using plot () function, Pie Chart, Scatter Plot, Histogram, Bar Chart).
29. Write programs making use of PuLp library.
30. Write programs making use of Scipy.

Essential Readings:

1. Deitel, P. J. (2019). Python Fundamentals. Pearson.
2. Dierbach, C. (2012). Introduction to computer science using python: a computational problem-solving focus. Wiley Publishing.
3. Guttag, J. V. (2013). Introduction to computation and programming using Python. MIT Press.
4. Lambert, K. A. (2018). Fundamentals of python: first programs. Cengage Learning.
5. Lutz, M., & Lutz, M. (1996). Programming python (volume 8). O'Reilly Media, Inc.
6. Thareja, R. (2017). Python programming using problem solving approach. Oxford University Press.
7. VanderPlas, J. (2016). Python data science handbook: essential tools for working with data. O'Reilly Media, Inc.

Suggested Readings: Nil

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

Discipline Specific Elective – Semester II

DSE - 2(a): Applied Multivariate Analysis

DSE - 2(b): Financial Management

DSE - 2(c): Fundamentals of Managerial Economics

DSE - 2(d): Marketing Research

DSE - 2(e): Quality Management

DSE - 2(f): Soft Computing

DISCIPLINE SPECIFIC ELECTIVE
DSE-2(a): APPLIED MULTIVARIATE ANALYSIS

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Applied Multivariate Analysis (DSE - 2(a))	**4**	**3**	**1**	**0**	**-**	Graduation

Learning Objectives:

The learning objectives of this course are as follows:
- Understand the theoretical foundations of multivariate statistical methods.
- To teach the application of data reduction techniques such as Principal Component Analysis (PCA) and Factor Analysis.
- To teach about classification methods, including Discriminant Analysis and Logistic Regression.
- Fit and interpret Multivariate Regression Models and Canonical Correlation Analysis.
- Visualize and interpret multivariate data using appropriate methods.

Learning Outcomes:

Students completing this course will be able to:
- Understand the role of multivariate techniques in strategic decision-making
- Have an understanding of statistical methods to analyse datasets with multiple variables simultaneously.
- Choose appropriate multivariate methods based on the research question.
- Create Interpretation of multivariate analysis results in a practical context.
- Understand different statistical data analysis techniques that are used for managerial decisions
- Apply Multivariate techniques in marketing, finance, healthcare, and social sciences.
- Have an understanding of Model selection and validation on real-world datasets.
- Make data-driven decisions from multivariate analysis.

Syllabus of DSE-2(a):

Unit I: Introduction to Multivariate Techniques **(9 hours)**
Range of Multivariate techniques available, selecting a multivariate technique, Use of Multivariate analysis methods across a range of industries and in Research & development.

Unit II: **Dependency Techniques** **(9 hours)**
Multiple Linear Regression Models, Logistic Regression & Receiver Operating Characteristic (ROC) curve, Discriminant Analysis.

Unit III: Interdependency Techniques (13 hours)
Correspondence Analysis, Conjoint Analysis, Chi-square Automatic Interaction Detector (CHAID) Analysis, Interpretive Structural Modeling (ISM), Matriced' Impacts Croise's Multiplication Appliquée a UN Classement (MICMAC) analysis. Exploratory and Confirmatory Factor Analysis, Application of the Henry Garrett Technique and other related interdependent techniques.

Unit IV: Some Assessment Techniques for Managerial Discretion (14 hours)
Application of Multi-Attribute Utility Theory, Two-way Assessment Methods, Understanding the importance of Normalization and some ranking techniques for managerial decision making**,** The Kano Model (to explore and measure customer needs), and Introduction to Market Basket Analysis.

Tutorial component (if any) - Yes (15 hours)

Practical component (if any) - Nil

Essential Readings:

1. Tzeng, G.H, Huang, J.J, (2011). Multiple Attribute Decision making Methods and Applications. CRC Press, Taylor & Francis Group
2. Backhaus, K., Erichson, B., Gensler, S., Weiber, R., & Weiber, T. (2021). Multivariate analysis. Springer Books, 10(1), 973-8.
3. Johnson, Richard A.; Wichern, Dean W. (2007). Applied Multivariate Statistical Analysis (Sixth ed.). Prentice Hall.
4. Johnson R A; Wichern D, (2002). Applied Multivariate Statistical Analysis: Prentice-Hall, New Jersey.
5. Johnson, D. E. (1998). Applied Multivariate Methods for Data Analysts: Duxbury Press.

Suggested Readings:

1. Sheth, J. N. (2011). What is Multivariate Analysis?. Marketing Classics Press.
2. Sharma, S. (1996) Applied Multivariate Techniques. Wiley

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

DISCIPLINE SPECIFIC ELECTIVE
DSE-2(b): FINANCIAL MANAGEMENT

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Financial Management (DSE-2(b))	**4**	**3**	**1**	**0**	**-**	Graduation

Learning Objectives:

- This introductory course in financial management focuses on the practical aspects of corporate finance.
- The course emphasizes understanding of financial management and working knowledge of the financial environment in which the firm operates to develop appropriate financial strategies.
- The course also covers the application of optimization techniques to financial management problems.

Learning Outcomes:

Students completing this course will be able to:
- Identify the objective of the firm's managerial finance role and outline the implications of the separation of ownership and control.
- Evaluate financial statements using ratio analysis and apply various techniques based on the time value of money. Explain general concepts of valuing financial assets and calculate the value of debt and equity securities.
- Understand the risk-return trade-off and demonstrate risk measurement through the Capital Asset Pricing Model (CAPM).
- Apply various capital budgeting techniques and calculate the weighted average cost of capital (WACC).
- Outline alternative sources of long-term funds and contrast operating leverage and financial leverage. Identify appropriate Operational Research techniques for financial decision making.

Syllabus of DSE-2(b):

Unit I: Introduction to Financial Management (14 hours)
Meaning, nature and scope of financial management. Financial markets. Financial Management goal: profit vs. wealth maximization. Finance functions: investment, financing and dividend decisions. Time Value of Money: Future and Present Value; Ordinary Annuity, Annuity Due and Perpetuity, Effective Annual Interest Rate (EAR), Loan Amortization. Valuation of Securities: Bonds and their Valuation, Bond Yields. Common and Preferred Stocks and their Valuation. Relationship (Trade-off) between risk and return, Capital Asset pricing model. Financial Statements: Balance Sheet, Income Statement, Statement of Cash Flows. Analysis of Financial Statements: Ratio Analysis, Du Pont Equations. Financial Planning and Forecasting: percentage of sales method, AFN Equation, cash budget.

Unit II: Capital Budgeting (10 hours)
Capital Budgeting process, Project Selection. Estimation of project cash flows, Capital Budgeting Techniques: Payback Period Methods, Average rate of return, Net Present Value methods, IRR, Benefit-Cost ratio, Capital rationing. Cost of Capital: Meaning and significance of cost of capital, Cost of debt, Cost of Equity and reserves, Cost of preferred stock, weighted average cost of capital, Factors affecting cost of capital.

Unit III: Operating and Financial Leverage (10 hours)
Measurement of leverages; Effects of operating and financial leverage on profit, analyzing alternate financial Plans, combined leverage. Capital Structure: Introduction, Factors affecting capital structure, Features of an optimal capital structure, Miller Modigliani propositions I and
II. Dividend policy: Aspects of dividend policy, practical consideration, forms of dividend policy, share splits.

Unit IV: Working Capital Management (11 hours)
Concepts, needs, Determinants, issues and estimation of working capital, Accounts Receivables Management, Inventory management, Cash management. Financing of working capital, Source of working capital: Spontaneous Source and Negotiable Source, Types of bank finance: commercial papers, other sources. Analytical approach to finance: Application of Operational Research techniques to the problems in Financial Decision Making.

Tutorial component (if any) - Yes (15 hours)

Practical component (if any) - Nil

Essential Readings:

1. Brealey, R., Myres, S., Franklin, A., Edmans, A., & Mohanty, P. (2023). Principles of Corporate Finance (14th ed.). McGraw Hill (India).
2. Brigham, E. F., & Michael, C.E. (2017). Financial Management- Theory and Practice (15th ed.). Cengage (India).
3. Cornuejols, G., Peña, J., & Tütüncü, R. (2018). Optimization methods in finance. Cambridge University Press.
4. Keown, A. J., Martin, J. D., & Petty, J. W. (2020). Foundations of finance (10th ed.). New Jersey: Pearson.
5. Khan, M. Y., & Jain, P. K. (2018). Financial Management: Text, Problems and Cases (8th ed.). McGraw Hill (India).
6. Spronk, J. (1981). Interactive Multiple Goal Programming: Applications to Financial Planning. Martinus Nijhoff Publishing.
7. Van Horne, J. C., & Wachowicz, J. M. (2008). Fundamentals of Financial Management (13th ed.). Harlow: Prentice Hall Inc.

Suggested Readings: Nil

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

DISCIPLINE SPECIFIC ELECTIVE
DSE-2(c): FUNDAMENTALS OF MANAGERIAL ECONOMICS

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Fundamentals of Managerial Economics (DSE-2(c))	**4**	**3**	**1**	**0**	**-**	**Graduation**

Learning Objectives:

The learning objectives of this course are as follows:
- To familiarize students with the fundamental concepts and theories of Managerial Economics.
- To strengthen managerial decision-making skills in today's competitive environment.
- The course also covers the application of optimization techniques to managerial decision making.

Learning Outcomes:

Students completing this course will be able to:
- Develop an understanding of basic concepts and issues in Managerial Economics and their applications in business decisions.
- Apply theories and tools covered under economics for analyzing business environment.
- Identify different economic factors and their importance in managerial decision making.
- Understand, evaluate and forecast demand and production functions in economics.
- Design competitive strategies according to the market structure.
- Enhance their knowledge of how markets operate and the capability in making economic predictions about markets.
- Identify appropriate optimization techniques for managerial decision making.

Syllabus DSE-2(c):

Unit I: Introduction to Managerial Economics (8 hours)
Objectives and features of managerial economics, Scope and importance of managerial economics, Goals of managerial decisions, Managerial decision making as optimizing with constraints, Marginal analysis in decision making. Formulation of relationships among economic variables using regression analysis. Basic optimization techniques for economic analysis.

Unit II: Demand and Supply Analysis (15 hours)
Theory of Demand: determinants of demand; law of demand; price elasticity of demand; income elasticity of demand; cross-price elasticity of demand; demand forecasting, theory of consumer behavior, Preference, Utility functions, Indifference curve. Supply Analysis: determinants of supply; law of supply; elasticity of supply, Market equilibrium: determination of equilibrium price and quantity.

Unit III: Production and Cost Analysis (11 hours)
Theory of production: factors of production; production function; law of variable proportions; returns to scale, Profit maximization: Constrained optimization approach. Cost Analysis: concept of costs; short-run and long-run costs; average and marginal costs; total, fixed and variable costs.

Unit IV: Market Structure and Pricing Practices (11 hours)
Market Structure: Different forms of markets - Perfect competition, Duopoly, Monopoly, Monopolistic competition, Oligopoly. Price behavior in different market structures. Profit maximization under different market structures. Role of government in market economies.

Tutorial component (if any) - Yes (15 hours)

Practical component (if any) - Nil

Essential Readings:

1. Baye, M., & Prince, J. (2022). Managerial economics and business strategy (10th ed.). McGraw Hill.
2. Hirschey, M. (2013). Managerial economics (12th ed.). Cengage Learning.
3. Koutsoyiannis, A. (2008). Modern Microeconomics (2nd ed.). Palgrave, McMillan.
4. Mankiw, N. G. (2024). Principles of Microeconomics (10th ed.). Cengage Learning.
5. Salvator, D., & Rastogi, S.K. (2020). Managerial Economics: Principles and Worldwide Applications (9th ed.). Oxford University Press.
6. Thomas C. R., & Maurice, S.C. (2020). Managerial Economics: Foundations of Business Analysis and Strategy (12th ed.). McGraw Hill.

Suggested Readings: Nil

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

DISCIPLINE SPECIFIC ELECTIVE
DSE-2(d): MARKETING RESEARCH

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Marketing Research (DSE-2(d))	**4**	**3**	**1**	**0**	**-**	Graduation

Learning Objectives:

The learning objectives of this course are as follows:
- This course aims to give the students an in-depth understanding of marketing research and its role in strategic decision-making.
- The course introduces the concepts of marketing research process and research design.
- This course shall provide concepts for measurement, scaling, and sampling design.
- To teach students about data analysis techniques relevant to marketing research.

Learning Outcomes:

Students completing this course will be able to:
- Understand the role of marketing research in strategic decision-making.
- Identify various steps involved in the marketing research process.
- Develop the research objectives and identify the appropriate market research design.
- Manage the Data Collection process.
- Understand different statistical data analysis techniques that are used in marketing research.
- Interpret the data analysis results in the context of the marketing problem under study.

Syllabus of DSE-2(d):

Unit I: Understanding Marketing Research (5 hours)
Concept of marketing research and its objectives, Applications of marketing research, Defining the marketing research problem and developing an approach.

Unit II: Research Design Formulation (8 hours)
Planning, Research design classification, Potential sources of error, Exploratory and descriptive research, Experimental research.

Unit III: Methods of Data Collection (12 hours)
Primary and secondary data collection, Advantages and limitations of primary and secondary data, Measurement and scaling techniques, Questionnaire design process. Sampling: sampling design process, classification of sampling techniques, Non-probability and probability sampling techniques, and sample size determination.

Unit IV: Statistical Techniques for Data Analysis (20 hours)
Data processing: Testing of hypothesis, Analysis of variance and covariance, MANOVA, Discriminant Analysis, Factor Analysis, Cluster Analysis, Conjoint Analysis.

Tutorial component (if any) - Yes (15 hours)

Practical component (if any) - Nil

Essential Readings:

1. Aaker, D. A., Kumara, V., & Day, G. S. (2007). Marketing research. John Wiley & Sons Inc.
2. Green, P. E., Tull, D. S., & Album, G. (1999). Research for marketing decisions: Prentice Hall of India.
3. Hague, P. N., Hague, N., & Morgan, C. A. (2004). Market research in practice: a guide to the basics. Kogan Page Publishers.
4. Heeringa, S. G., West, B. T., & Berglund, P. A. (2017). Applied survey data analysis. Chapman and Hall.
5. Malhotra, N., Hall, J., Shaw, M., & Oppenheim, P. (2006). Marketing research: an applied orientation. Pearson Education.
6. Smith, S. M., & Albaum, G. S. (2005). Fundamentals of marketing research. SAGE Publications Inc.

Suggested Readings: Nil

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

DISCIPLINE SPECIFIC ELECTIVE
DSE-2(e): QUALITY MANAGEMENT

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Quality Management (DSE-2(e))	**4**	**3**	**1**	**0**	**-**	Graduation

Learning Objectives:

The learning objectives of this course are as follows:

- To impart the knowledge of concepts related to quality management.
- To tackle the issues and problems and to develop practical skills for continuous quality improvement.

Learning Outcomes:

Students completing this course will be able to:

- Define quality management concepts, its evolution, and key quality gurus with their contributions.
- Explain statistical concepts like sampling, OC curve and Quality 4.0 for digital transformation.
- Implement SPC tools (CUSUM, EWMA), Six Sigma (DMAIC, DMADV), and defect prevention methods like Poka-yoke and QFD.
- Identify causes of process variation, compare quality improvement techniques, and evaluate benchmarking and Kaizen strategies.
- Assess Six Sigma effectiveness, analyze statistical control tools, and examine the role of Quality 4.0 in modern industries.

Syllabus of DSE-2(e):

Unit I: Evolution and Frameworks of Quality Management (10 hours)
Evolution of quality management, Concepts of product and service quality, Dimensions of quality, Major quality gurus: Deming, Ishikawa, Taguchi; Quality costs, Total quality management, Total quality management excellence model, Quality 4.0: Digital Transformation in Quality.

Unit II: Statistical Process Control and Quality Assurance (12 hours)
Process and product quality, Causes of variations (assignable and unassignable), Statistical process control, Process control charts: variable control charts (X-bar and R, X-bar and S) and attribute control charts (np and p, c and u), CUSUM & EWMA Charts for Early Defect Detection Sampling, Sampling distribution, Acceptance sampling plan: single, double and sequential; Acceptable quality level, Average outgoing quality, Average outgoing quality limit, Operating characteristic curve.

Unit III: Quality Tools and Process Capability (12 hours)
Pareto chart, Cause and effect diagram, Check sheet, Histogram, Scatter diagram, Process control charts applications, Graphs: circle graph, bar graph and radar graph. Affinity diagram, Relations diagram, Systematic or tree diagram, Matrix diagram, Matrix data analysis, Arrow diagram, Process capability indices.

Unit IV: Six Sigma, Continuous Improvement and Lean Quality Management (11 hours)
Concept of six sigma, Implementation of six sigma: DMAIC, DMADV, Failure mode and effect analysis. Benchmarking for process/service improvement, Concepts of kaizen, Kaizen in practice, Kaizen versus innovation, Lean, 5S, Quality function deployment, Quality control circle, Poka- Yoke.

Tutorial component (if any) - Yes (15 hours)

Practical component (if any) - Nil

Essential Readings:

1. Besterfield, D. H. (2004). Quality control. India: Pearson Education India.
2. Charantimath, P.M. (2011). Total quality management. India: Pearson Education India.
3. Evans, J. R., & Lindsay, W. M. (2002). The management and control of quality (volume 1). Cincinnati, OH: South-Western.
4. Montgomery, D. C. (2009). Introduction to statistical quality control. New York: John Wiley & Sons.
5. Gupta, S. C., & Kapoor, V. K. (2009). Fundamentals of applied statistics. India: Sultan Chand & Sons.

Suggested Readings: Nil

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

DISCIPLINE SPECIFIC ELECTIVE
DSE-2(f): SOFT COMPUTING

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Soft Computing (DSE-2(f))	**4**	**3**	**1**	**0**	**-**	Graduation

Learning Objectives:

The learning objectives of this course are as follows:
- Understand the foundational concepts of soft computing, including interval uncertainty, fuzzy logic, neural networks, and genetic algorithms, and their significance in modern computing.
- Analyze and apply soft computing techniques to real-world problems in business, finance, artificial intelligence, and optimization.

Learning Outcomes:

Students completing this course will be able to:
- Understand principles and differences between soft and traditional computing techniques.
- Apply interval analysis and fuzzy logic concepts to create decision-making models for business and finance scenarios.
- Design, train, and implement neural network architectures and genetic algorithms.
- Tackle classification, prediction, and optimization problems using soft computing techniques.

Syllabus of DSE-2(f):

Unit I: Interval Uncertainty (10 hours)
Definition and importance of soft computing. Comparison with traditional computing. Definition and representation of intervals. Arithmetic operations on intervals. Properties of interval numbers. Interval uncertainty in business, finance, and optimization applications.

Unit II: Fuzzy Sets and Logic (11 hours)
Introduction to fuzzy sets and fuzzy logic. Operations on fuzzy sets. Membership functions. Defuzzification process. Fuzzy relations, rules, and implications. Applications of fuzzy logic.

Unit III: Neural Networks (12 hours)
Neural networks: Fundamentals, Single and multiple input neurons, Layers, and Activation functions. Single-layer and multi-layer perceptrons. Types of neural networks: feedforward, and recurrent. Supervised vs. unsupervised learning. Learning algorithm: Gradient descent method. Applications in function approximation, prediction, and classification.

Unit IV: Genetic Algorithms **(12 hours)**
Introduction to Genetic Algorithms. Basic concepts: Encoding methods, Chromosomes, Population, Fitness Function. Operators: Inversion, Selection, Crossover, and Mutation. Applications in optimization, finance, and artificial intelligence.

Tutorial component (if any) - Yes **(15 hours)**

Practical component (if any) - Nil

Essential Readings:

1. Rajasekaran, S. & Vijayalakshmi Pai, S. A. (2007). Neural Networks, Fuzzy Logic and Genetic Algorithms: Synthesis, and Applications. Prentice Hall of India.
2. Goldberg D. E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Addison Wesley
3. Moore, R. E., Kearfott, R. B. & Cloud, M. L. (2009). Introduction to Interval Analysis. Society for Industrial and Applied Mathematics, United States.

Suggested Readings: Nil

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

Skill Enhancement Course – Semester II

SEC - 2: Spreadsheet and Data Visualization

SKILL ENHANCEMENT COURSE
SEC-2: SPREADSHEET MODELING AND DATA VISUALIZATION

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Spreadsheet Modeling and Data Visualization (SEC-2)	**2**	**1**	**0**	**1**	**-**	Graduation

Learning Objectives:

- To equip students with essential spreadsheet skills for data management, cleaning, visualization, and analysis.
- The course will help students to apply techniques to real-world business analytics and operations research problems.

Learning Outcomes:

Students completing this course will be able to:
- Understand spreadsheet functionalities, including file security, referencing, and formula applications.
- Apply data cleaning techniques to handle missing values, errors, outliers, and text formatting for high-quality datasets.
- Utilize data visualization principles to create meaningful charts, dashboards, and visual comparisons of structured data.
- Perform data analysis using consolidation techniques like what-if analysis, scenario manager, and solver for decision-making.
- Construct and manipulate pivot tables to analyse large datasets, create calculated fields, and generate reports effectively.
- Implement spreadsheet-based analytical solutions in business and operations research applications.

Syllabus of SEC-2:

Unit I: Fundamentals of Spreadsheets and Data Cleaning*

(3 Theory hours + 6 Practical hours)

Basic concepts of Spreadsheets, implementing file level security and protecting data within the worksheet; Understanding absolute, relative and mixed referencing in formulas, referencing cells in other worksheets and workbooks. Data cleaning and processing: Handling Blank Cells & Duplicates, Text Cleaning & Formatting, Finding & Replacing Data: Advance Find & Replace, Handling Errors & Missing Data, Working with Named Ranges, Handling Outliers & Data Validation. Working with inbuilt function categories like mathematical, statistical, text and date functions.

Unit II: Data Visualization: Principles and Techniques*

(4 Theory hours + 8 Practical hours)

Data Visualization: Need, Importance and Usage. Seven Stages of Visualizing Data, Types of Charts. Tabular versus Visual Data Analysis. Dashboards, Visualization of Structured Data: Charts on Univariate, Charts on Visualizing Multiple Measures.

Unit III: **Data Consolidation, Analysis, and Visualization Strategies***

(4 Theory hours + 8 Practical hours)

Data consolidation commands: Performing what-if analysis: Types of what if analysis (manual, data tables, scenario manager), what-if analysis in reverse (goal-seek, solver), Choosing a chart type, understanding data points and data series, editing and formatting chart elements, and creating sparkline graphics.

Unit IV: Data Analysis with Pivot Tables and Applications*

(4 Theory hours + 8 Practical hours)

Constructing a Cross-Tabulation, Analyzing data using pivot tables: Creating, formatting and modifying a pivot table, sorting, filtering and grouping items, creating calculated field and calculated item, creating pivot table charts, producing a report with pivot tables.

*Case Studies and Real-World Applications: Business Analytics and Operations Research Applications.

Essential Readings:

1. Knaflic, C. N. (2015). Storytelling with data: A data visualization guide for business professionals. John Wiley & Sons.
2. Ragsdale, C. T. (2017). Spreadsheet modeling & decision analysis. Thomson south-western.
3. Evans, J. R. (2019). Business analytics: Methods, models, and decisions. Pearson.
4. Evergreen, S. D. (2019). Effective data visualization: The right chart for the right data. SAGE publications.
5. McFedries, P. (2022). Excel Formulas and Functions. Pearson Education.
6. Winston, W. (2021). Microsoft Excel Data Analysis and Business Modeling (Office 2021 and Microsoft 365). Microsoft Press.

Suggested Readings: Nil

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

Generic Elective - Semester II

GE - 2(a): Queueing Theory

GE - 2(b): Marketing Research

GE - 2(c): Quality Management

GE - 2(d): Soft Computing

GENERIC ELECTIVE
GE-2(a): QUEUEING THEORY

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Queueing Theory (GE-2(a))	**4**	**3**	**1**	**0**	**-**	Graduation

Learning Objectives:

The learning objectives of this course are as follows:
- To impart knowledge of queueing theory and provide the students with a rigorous framework which would enable them to model and analyze queueing systems.
- To provide the students necessary mathematical support and confidence to the students to tackle real life problems.
- To explore both theory and application of fundamental and advanced models in this field.

Learning Outcomes:

Students completing this course will be able to:
- Gain a deep understanding of the theoretical background of queueing systems.
- Understand and compute quantitative metrics of performance for queueing systems.
- Apply and extend queueing models to analyze real world systems.

Syllabus of GE-2(a):

Unit I: Introduction (10 hours)
Basic Concepts of Stochastic Process, Markov Chains, Introduction to Queueing Systems, Characteristics of Queueing Systems, Counting Process, Poisson Process (Pure-Birth Process and Pure-Death Process), Expected Measures of System Performance, Queueing Simulation: Data Generation and Book-Keeping.

Unit II: Markovian Queueing Models (15 hours)
General Birth-Death Process, Infinite Capacity Queues: Single-Server Queues (M/M/1) and Multi-Server Queues (M/M/c), Finite Capacity Queues: M/M/1/K, M/M/c/K and M/M/c/c (Erlang's Loss System), Queues with Unlimited Service (M/M/∞), Finite-Source Queues, Queues with State-Dependent Service, Queues with Impatience (M/M/1 Balking and M/M/1 Reneging).

Unit III: Advanced Markovian Queueing Models (15 hours)
Queues with Bulk Arrival ($M^{[X]}$/M/1), Queues with Bulk Service (M/$M^{[Y]}$/1), Erlangian Queueing Models (M/E_k/1 and E_k/M/1), Concept of Imbedded Markov Chains, Queues with General Arrivals (G/M/1), Queues with General Service (M/G/1).

Unit IV: Decision Problems in Queueing Systems (5 hours)
Design and Control Problems in Queueing Systems, Queueing System with Server Vacation, Queueing System with Removable Server.

Tutorial component (if any) - Yes (15 hours)

Practical component (if any) - Nil

Essential Readings:

1. Bhat, U. N. (2015). An introduction to Queueing Theory: Modelling and Analysis in Applications (Statistics for Industry and Technology) (2^{nd} Edition). Birkhauser Boston.
2. Cooper, R.B. (1981). Introduction to Queueing Theory (2^{nd} Edition). Elsevier North Holland.
3. Cox, D. R. and Smith, W. L. (1991). Queues. Chapman and Hall/CRC.
4. Gross, Donald, Shortle , John F., Thompson, James M., and Harris, Carl M. (2018). Fundamentals of Queueing Theory (5^{th} Edition), John Wiley and Sons Inc. Pte. Ltd.
5. Kleinrock L. (1975). Queueing Systems, Volume 1: Theory, John Wiley.
6. Medhi, J. (2003). Stochastic Models in Queueing Theory (2^{nd} Edition), Academic Press.
7. Prabhu, N. U. (2012). Foundations of Queueing Theory (International Series in Operations Research & Management Science), Springer (Softcover reprint of the original 1st ed. 1997 edition.)
8. Satty, T. L. (1983). Elements of Queueing Theory with Applications, Dover Publications, NY. (Reprint Edition).

Suggested Readings: Nil

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

GENERIC ELECTIVE
GE-2(b): MARKETING RESEARCH

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Marketing Research (GE-2(b))	**4**	**3**	**1**	**0**	**-**	Graduation

Learning Objectives:

The learning objectives of this course are as follows:
- This course aims to give the students an in-depth understanding of marketing research and its role in strategic decision-making.
- The course introduces the concepts of marketing research process and research design.
- This course shall provide concepts for measurement, scaling, and sampling design.
- To teach students about data analysis techniques relevant to marketing research.

Learning Outcomes:

Students completing this course will be able to:
- Understand the role of marketing research in strategic decision-making.
- Identify various steps involved in the marketing research process.
- Develop the research objectives and identify the appropriate market research design.
- Manage the Data Collection process.
- Understand different statistical data analysis techniques that are used in marketing research.
- Interpret the data analysis results in the context of the marketing problem under study.

Syllabus of GE-2(b):

Unit I: Understanding Marketing Research (5 hours)
Concept of marketing research and its objectives, Applications of marketing research, Defining the marketing research problem and developing an approach.

Unit II: Research Design Formulation (8 hours)
Planning, Research design classification, Potential sources of error, Exploratory and descriptive research, Experimental research.

Unit III: Methods of Data Collection (12 hours)
Primary and secondary data collection, Advantages and limitations of primary and secondary data, Measurement and scaling techniques, Questionnaire design process. Sampling: sampling design process, classification of sampling techniques, Non-probability and probability sampling techniques, and sample size determination.

Unit IV: Statistical Techniques for Data Analysis **(20 hours)**
Data processing: Testing of hypothesis, Analysis of variance and covariance, MANOVA, Discriminant Analysis, Factor Analysis, Cluster Analysis, Conjoint Analysis.

Tutorial component (if any) - Yes **(15 hours)**

Practical component (if any) - Nil

Essential Readings:

1. Aaker, D. A., Kumara, V., & Day, G. S. (2007). Marketing research. John Wiley & Sons Inc.
2. Green, P. E., Tull, D. S., & Album, G. (1999). Research for marketing decisions: Prentice Hall of India.
3. Hague, P. N., Hague, N., & Morgan, C. A. (2004). Market research in practice: a guide to the basics. Kogan Page Publishers.
4. Heeringa, S. G., West, B. T., & Berglund, P. A. (2017). Applied survey data analysis. Chapman and Hall.
5. Malhotra, N., Hall, J., Shaw, M., & Oppenheim, P. (2006). Marketing research: an applied orientation. Pearson Education.
6. Smith, S. M., & Albaum, G. S. (2005). Fundamentals of marketing research. SAGE Publications Inc.

Suggested Readings: Nil

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

GENERIC ELECTIVE
GE-2(c): QUALITY MANAGEMENT

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Quality Management (GE-2(c))	**4**	**3**	**1**	**0**	**-**	Graduation

Learning Objectives:

The learning objectives of this course are as follows:

- To impart the knowledge of concepts related to quality management.
- To tackle the issues and problems and to develop practical skills for continuous quality improvement.

Learning Outcomes:

Students completing this course will be able to:

- Define quality management concepts, its evolution, and key quality gurus with their contributions.
- Explain statistical concepts like sampling, OC curve and Quality 4.0 for digital transformation.
- Implement SPC tools (CUSUM, EWMA), Six Sigma (DMAIC, DMADV), and defect prevention methods like Poka-yoke and QFD.
- Identify causes of process variation, compare quality improvement techniques, and evaluate benchmarking and Kaizen strategies.
- Assess Six Sigma effectiveness, analyze statistical control tools, and examine the role of Quality 4.0 in modern industries.

Syllabus of GE-2(c):

Unit I: Evolution and Frameworks of Quality Management (10 hours)
Evolution of quality management, Concepts of product and service quality, Dimensions of quality, Major quality gurus: Deming, Ishikawa, Taguchi; Quality costs, Total quality management, Total quality management excellence model, Quality 4.0: Digital Transformation in Quality.

Unit II: Statistical Process Control and Quality Assurance (12 hours)
Process and product quality, Causes of variations (assignable and unassignable), Statistical process control, Process control charts: variable control charts (X-bar and R, X-bar and S) and attribute control charts (np and p, c and u), CUSUM & EWMA Charts for Early Defect Detection Sampling, Sampling distribution, Acceptance sampling plan: single, double and sequential; Acceptable quality level, Average outgoing quality, Average outgoing quality limit, Operating characteristic curve.

Unit III: Quality Tools and Process Capability (12 hours)
Pareto chart, Cause and effect diagram, Check sheet, Histogram, Scatter diagram, Process control charts applications, Graphs: circle graph, bar graph and radar graph. Affinity diagram, Relations diagram, Systematic or tree diagram, Matrix diagram, Matrix data analysis, Arrow diagram, Process capability indices.

Unit IV: Six Sigma, Continuous Improvement and Lean Quality Management (11 hours)
Concept of six sigma, Implementation of six sigma: DMAIC, DMADV, Failure mode and effect analysis. Benchmarking for process/service improvement, Concepts of kaizen, Kaizen in practice, Kaizen versus innovation, Lean, 5S, Quality function deployment, Quality control circle, Poka- Yoke.

Tutorial component (if any) - Yes (15 hours)

Practical component (if any) - Nil

Essential Readings:

1. Besterfield, D. H. (2004). Quality control. India: Pearson Education India.
2. Charantimath, P.M. (2011). Total quality management. India: Pearson Education India.
3. Evans, J. R., & Lindsay, W. M. (2002). The management and control of quality (volume 1). Cincinnati, OH: South-Western.
4. Montgomery, D. C. (2009). Introduction to statistical quality control. New York: John Wiley & Sons.
5. Gupta, S. C., & Kapoor, V. K. (2009). Fundamentals of applied statistics. India: Sultan Chand & Sons.

Suggested Readings: Nil

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

GENERIC ELECTIVE
GE-2(d): SOFT COMPUTING

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Soft Computing (GE-2(d))	**4**	**3**	**1**	**0**	**-**	Graduation

Learning Objectives:

The learning objectives of this course are as follows:
- Understand the foundational concepts of soft computing, including interval uncertainty, fuzzy logic, neural networks, and genetic algorithms, and their significance in modern computing.
- Analyze and apply soft computing techniques to real-world problems in business, finance, artificial intelligence, and optimization.

Learning Outcomes:

Students completing this course will be able to:
- Understand principles and differences between soft and traditional computing techniques.
- Apply interval analysis and fuzzy logic concepts to create decision-making models for business and finance scenarios.
- Design, train, and implement neural network architectures and genetic algorithms.
- Tackle classification, prediction, and optimization problems using soft computing techniques.

Syllabus of DSE-2(f):

Unit I: Interval Uncertainty (10 hours)
Definition and importance of soft computing. Comparison with traditional computing. Definition and representation of intervals. Arithmetic operations on intervals. Properties of interval numbers. Interval uncertainty in business, finance, and optimization applications.

Unit II: Fuzzy Sets and Logic (11 hours)
Introduction to fuzzy sets and fuzzy logic. Operations on fuzzy sets. Membership functions. Defuzzification process. Fuzzy relations, rules, and implications. Applications of fuzzy logic.

Unit III: Neural Networks (12 hours)
Neural networks: Fundamentals, Single and multiple input neurons, Layers, and Activation functions. Single-layer and multi-layer perceptrons. Types of neural networks: feedforward, and recurrent. Supervised vs. unsupervised learning. Learning algorithm: Gradient descent method. Applications in function approximation, prediction, and classification.

Unit IV: Genetic Algorithms **(12 hours)**
Introduction to Genetic Algorithms. Basic concepts: Encoding methods, Chromosomes, Population, Fitness Function. Operators: Inversion, Selection, Crossover, and Mutation. Applications in optimization, finance, and artificial intelligence.

Tutorial component (if any) - Yes **(15 hours)**

Practical component (if any) - Nil

Essential Readings:

1. Rajasekaran, S. & Vijayalakshmi Pai, S. A. (2007). Neural Networks, Fuzzy Logic and Genetic Algorithms: Synthesis, and Applications. Prentice Hall of India.
2. Goldberg D. E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Addison Wesley
3. Moore, R. E., Kearfott, R. B. & Cloud, M. L. (2009). Introduction to Interval Analysis. Society for Industrial and Applied Mathematics, United States.

Suggested Readings: Nil

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

Annexure-4

UNIVERSITY OF DELHI

DEPARTMENT OF STATISTICS

1st year of 2-year PG Program under PGCF-Level 6

Proposed Syllabus
(Effective from AY 2025-26)

M.A./M.Sc. Statistics Programme Details:

Programme Structure:

Two Year M.A./M.Sc. Statistics programme is a course divided into 2+2 semesters. A student is required to complete minimum **22** credits for completion of each semester.

		Semester	Semester	Level
Part – I	First Year	Semester I	Semester II	6
Part – II	Second Year	Semester III	Semester IV	6.5

Course Credit Scheme:

In each semester 3 Core and either 2DSE or 1DSE+1GE are required.

Semester	Discipline-Specific Core (DSC)		Discipline-Specific Elective (DSE)		Generic Elective (GE)		Skill Based/Specialized Laboratory		Total Credits
	No. of Papers	Credits	No. of Papers	Credits	No. of Papers	Credits	No. of Papers	Credits	
I	03	12	02	8	00	0	1	2	**22**
			01	4	01	4			
II	03	12	02	8	00	0	1	2	**22**
			01	4	01	4			
Total Credits for Ist Year of Two-Year Courses									**44**

Semester Wise Details:

Semester –I					
Discipline-Specific Core Courses: 3					
Course code	**Course Title**	**Credits in each course**			
		Theory	Tutorial	Practical	Credits
1a	Probability Theory	3	1	0	4
1b	Statistical Methodology	3	1	0	4
1c	Survey Sampling	3	0	1	4
Total credits		**9**	**2**	**1**	**12**

Discipline-Specific Elective (DSE) Courses:					
Course code	**Course Title**	**Credits in each course**			
		Theory	Tutorial	Practical	Credits

1a	Analysis	3	1	0	4
1b	Time Series Analysis	3	0	1	4
1c	Biostatistics	3	0	1	4
1d	Official and National Development Statistics	3	1	0	4
Total credits		**12**	**2**	**2**	**16**

Generic Elective (GE) Courses:					
Course code	**Course Title**	**Credits in each course**			
		Theory	Tutorial	Practical	Credits
1a	Statistical Computing using R	3	0	1	4
Total credits		**3**	**0**	**1**	**4**

Skill Based/Specialized Laboratory Courses:					
Course code	**Course Title**	**Credits in each course**			
		Theory	Tutorial	Practical	Credits
1a	Data Analysis using Excel	0	0	2	2
Total credits		**0**	**0**	**2**	**2**

Semester –II

Discipline-Specific Core (DSC) Core Courses: 3					
Course code	**Course Title**	**Credits in each course**			
		Theory	Tutorial	Practical	Credits
2a	Statistical Inference	3	1	0	4
2b	Design of Experiments	3	0	1	4
2c	Stochastic Processes	3	1	0	4
Total credits		**9**	**2**	**1**	**12**

Discipline-Specific Elective (DSE) Courses:					
Course code	**Course Title**	**Credits in each course**			
		Theory	Tutorial	Practical	Credits
2a	Linear Algebra	3	1	0	4
2b	Non-Parametric Inference	3	0	1	4
2c	Statistical Quality Control	3	0	1	4
2d	Reliability Theory	3	1	0	4
2e	Computational Techniques	2	0	2	4

Total credits	**14**	**2**	**4**	**20**

Generic Elective (GE) Courses:					
Course code	**Course Title**	**Credits in each course**			
		Theory	Tutorial	Practical	Credits
2a	Statistics for Research and Management Studies	2	0	2	4
Total credits		**2**	**0**	**2**	**4**

Skill Based/Specialized Laboratory Courses:					
Course code	**Course Title**	**Credits in each course**			
		Theory	Tutorial	Practical	Credits
1a	Data Analysis using Python	0	0	2	2
Total credits		**0**	**0**	**2**	**2**

First Year of Two-Year PG (M.A./ M.Sc. Statistics) Programme

Semester- I

Discipline-Specific Core (DSC) Course- 1a: Probability Theory

Course Title & Code	Credits	Credit Distribution of the Course		
		Lecture (45 Hours)	Tutorial (15 Hours)	Practical/ Practice
DSC 1a: Probability Theory	4	3	1	0

Course Objectives: The aim of the course is to pay a special attention to applications of measure theory in the probability theory, understanding of Week Law of Large Numbers, Strong Law of Large Numbers and the Central Limit Theorem with their applications.

Course Learning Outcomes: After successfully completing this course, students will be able to apply:

1. Concepts of random variables, sigma-fields, probability distributions, and the independence of random variables related to measurable functions.
2. Skills in working with measurable functions and sequences of random variables.
3. The weak laws of large numbers in practical scenarios.
4. The strong laws of large numbers to solve real-world problems.
5. The central limit theorem in data analysis and interpretation.
6. Principles of convergence and modes of convergence to assess statistical data.
7. Characteristic functions, as well as the uniqueness, inversion, and Levy continuity theorems, to advanced probability problems.

Unit I: (10 Hours)

Classes of sets, fields, σ -fields, minimal σ -field, Borel σ – field in R^K, sequence of sets, limsup and liminf of a sequence of sets. Measure, Probability measure, properties of a measure, Caratheodory extension theorem (statement only), Lebesgue measures on R^K**.**

Unit II: (11 Hours)

Measurable functions, Random variables, sequence of random variables, Integration of a measurable function with respect to a measure. Monotone convergence theorem, Fatou's lemma, Dominated convergence theorem. Characteristic functions, uniqueness/inversion/Levy continuity theorems.

Unit III: (12 Hours)

Markov's, Chebychev's and Kolmogorov's inequalities, Modes of stochastic convergence, Jenson, Liapounov, holder's and Minkowsky's inequalities, Sequence of random variables and modes of convergence (convergence in distribution, in probability, almost surely, and quadratic mean) and their interrelations. Statement of Slutsky's theorem, Borel –Cantelli lemma and Borel 0-1 law.

Unit IV: (12 Hours)

Concept of Independence, Laws of large numbers, Chebyshev's and Khinchine's WLLN, necessary and sufficient condition for the WLLN, strong law of large numbers and Kolmogorov's theorem, Central limit theorem, Lindeberg and Levy and Liapunov forms of CLT.

Suggested Readings:

1. Ash, R. B. and Doléans-Dade, C.A. (1999). Probability and Measure Theory, Second Edition, Academic Press, New York.
2. Bhat, B.R. (1999). Modern Probability Theory, 3rd Edition, New Age International Publishers.
3. Billingsley, P. (2012). Probability and Measure, Anniversery Edition, John Wiley & Sons.
4. Capinski, M. and Zastawniah (2001). Probability through problems, Springer.
5. Chung, K. L. (1974). A Course in Probability Theory, 2nd Edition, Academic Press, New York.

6. Feller, W. (1968). An Introduction to Probability Theory and its Applications, Vol. 1, 3rd Edition, John Wiley & Sons.
7. Parzen, E. (1960). Modern Probability Theory and its Application. Wiley Eastern Private Ltd.

Teaching Plan:

Week 1:	Classes of sets, fields, σ -fields, minimal σ -field, Borel σ – field in R^K, sequence of sets, limsup and liminf of a sequence of sets.
Week 2:	Measure, Probability measure, properties of a measure, Caratheodory extension theorem (statement only), Lebesgue measures on R^K
Week 3:	Measurable functions, Random variables, sequence of random variables.
Week 4:	Integration of a measurable function with respect to a measure. Monotone convergence theorem.
Week 5:	Fatou's lemma, Dominated convergence theorem.
Week 6:	Characteristic functions, uniqueness/inversion/Levy continuity theorems.
Week 7:	Markov's, Chebychev's and Kolmogorov's inequalities. Modes of stochastic convergence. Jensen, Liapunov, Holder's and Minkowsky's inequalities.
Week 8:	Measurable functions, Random variables Sequence of random variables
Week 9:	Modes of convergence (convergence in distribution, in probability, almost surely, and quadratic mean) and their interrelations.
Week 10:	Statement of Slutsky's theorem. Borel –Cantelli lemma and Borel 0-1 law.
Week 11:	Concept of Independence, Laws of large numbers, Chebyshev's
Week 12:	Khinchine's WLLN, necessary and sufficient condition for the WLLN. Strong
Week 13:	Law of large numbers and Kolmogorov's theorem. Strong law of large numbers and Kolmogorov's theorem.
Week 14:	Central limit theorem, Lindeberg and Levy and Liapunov forms of CLT.

Discipline-Specific Core (DSC) Course- 1b: Statistical Methodology

Course Title & Code	Credits	Credit Distribution of the Course		
		Lecture (45 Hours)	Tutorial (15 Hours)	Practical/ Practice
DSC 1b: Statistical Methodology	4	3	1	0

Course Objective: The aim of this course is to provide a thorough theoretical grounding in different type of distributions, non-central distributions, censoring, delta method, robust procedures etc.

Course Learning Outcomes:

After successful completion of this course, student will be able to:

1. Formulate the mathematical/statistical models for real data sets arising in various fields in order to analyse in respect of various useful characteristics of the populations.
2. Understand how to use non-central distributions in real life problems.
3. Understand different types of censoring schemes and their applications.
4. Work with incomplete data which is a challenging problem in today's life.

Unit I: (10 Hours)

Brief review of basic distribution theory, Symmetric distributions, Truncated distributions, Compound distributions, Mixture of distributions, Generalized power series distributions, Exponential family of distributions.

Unit II: (12 Hours)

Characterization of distributions (Geometric, negative exponential, normal, gamma), Non-central Chi-square, t and F distributions and their properties, Concept of censoring. Approximating distributions, Delta method and its applications, Approximating distributions of sample moments, Limiting moment generating function, Poisson approximation to negative binomial distribution.

Unit III: (12 Hours)

Order statistics-their distributions and properties. Joint and marginal distributions of order statistics. Extreme values and their asymptotic distributions (statement only) with applications. Tolerance intervals, coverage of ($X_{(r)}$, $X_{(s)}$). Generic theory of regression, fitting of polynomial regression by orthogonal methods, multiple regression, examination of regression equation.

Unit IV: (11 Hours)

Robust procedures, Robustness of sample mean, Sample standard deviation, Chi-square test and Student's t-test. Sample size determination for testing and estimation procedures (complete and censored data) for normal, exponential, Weibull and gamma distributions.

Suggested Readings:

1. Arnold, B.C., Balakrishnan, N., and Nagaraja, H.N. (1992). A First Course in Order Statistics, John Wiley & Sons.
2. Biswas, S. (1992). Topics in Statistical Methodology, Wiley-Blackwell.
3. David, H.A., and Nagaraja, H.N. (2003). Order Statistics, 3rd Edn., John Wiley & Sons.
4. Dudewicz, E.J. and Mishra, S.N. (1988). Modern Mathematical Statistics, Wiley, International Students' Edition.
5. Huber, P.J. (1981). Robust Statistics, John Wiley & Sons.
6. Johnson, N.L., Kotz, S. and Balakrishnan, N. (2000). Continuous Univariate Distributions, John Wiley & Sons.
7. Johnson, N.L., Kotz, S. and Balakrishnan, N. (2000). Discrete Univariate John Wiley & Sons.
8. Mukhopadhyay, P. (2015). Mathematical Statistics. New Central Book Agency.
9. Rao, C.R. (1973). Linear Statistical Inference and Its Applications, 2nd Edn., John Wiley & Sons.
10. Rohatgi, V.K. (1984). Statistical Inference, John Wiley & Sons.
11. Rohatgi, V.K. and Saleh, A. K. Md. E. (2005). An Introduction to Probability and Statistics, 2nd Edn., John Wiley & Sons.

Teaching Plan:

Week 1:	Concept of sample space, random variable, sigma field, minimal sigma field, probability measure, properties of probability density functions/probability mass functions and cumulative distribution functions, Concept of symmetric distributions.

Week 2:	Examples of discrete and continuous symmetric distributions, theorems based on symmetric distributions, Concept of truncated distributions, Examples of discrete and continuous truncated distributions, Concept of compound distributions, Examples of compound distributions.
Week 3:	Concept of Generalized power series distribution, moment generating function, Recurrence relation and cumulents of Generalized power series distribution, Particular cases of Generalized power series distribution.
Week 4:	Brief discussion of one parameter, two parameter and k-parameters exponential family of distributions. Particular cases of exponential family of distributions, Maximum likelihood estimation of exponential family of distributions and theorems based on exponential family of distributions.
Week 5:	Characterization properties and theorem of geometric distribution, Characterization properties of exponential distribution and related theorems, Characterization properties of normal distribution and related theorems. Characterization properties of gamma distribution.
Week 6:	Brief discussion of central Chi-square. Concept of Non-central Chi-square distribution, derivation of the probability density functions, characteristic functions, moment generating function, Cumulants and other theorems based on Non-central Chi-square distribution.
Week 7:	Brief discussion of t and F distributions. Concept of Non-central t and F distributions, derivation of their probability density functions, Derivation of r^{th} moment about origins of non-central t and F distributions. Derivation of mean and variance of non-central t and F distributions.
Week 8-9:	Discussion of Type I censoring, Type II censoring and Progressive censoring and problems based on these censoring schemes, Concept of Approximating distributions, First order, second order and higher order delta method. Problems based on delta method.
Week 10:	Approximating distributions of sample moments, Limiting moment generating function, Poisson approximation to negative binomial distribution, Concept of order statistics and problems/theorems based on order statistics.

Week 11:	Concept of tolerance intervals and problems based on tolerance intervals, Coverage of ($X_{(r)}$, $X_{(s)}$), Generic theory of linear and multiple regression.
Week 12:	Fitting of polynomial regression by orthogonal methods, multiple regression, and examination of regression equation.
Week 13:	Concept of Robust procedures, Robustness of sample mean, Sample standard deviation, Chi-square test and Student's t-test.
Week 14:	Sample size determination for testing and estimation procedures (complete and censored data) for normal, exponential, Weibull and gamma distributions.

Discipline-Specific Core (DSC) Course- 1c: Survey Sampling

Course Title & Code	Credits	Credit Distribution of the Course		
		Lecture (45 Hours)	Tutorial	Practical/ Practice (30 Hours)
DSC 1c: Survey Sampling	**4**	**3**	**0**	**1**

Course Objectives:

1. The main objective of this course is to learn techniques in survey sampling with practical applications in daily life which would be beneficial for the students to their further research.
2. To provide tools and techniques for selecting a sample of elements from a target population keeping in mind the objectives to be fulfilled and nature of population.

Course Learning Outcomes:

1. Understand the distinctive features of sampling schemes and its related estimation problems 2.
2. Learn about various approaches (design based and model-based) to estimate admissible parameters; with and without replacement sampling scheme, sampling with varying probability of selection.

3. Learn about the methods of post–stratification (stratified sampling) and controlled sampling and also double sampling procedure with unequal probability of selection.
4. Learn about the applications of sampling methods; systematic, stratified and cluster sampling.
5. Learn about the randomized response techniques.

Unit I: (10 Hours)

Basic ideas and distinctive features of sampling, Probability sampling designs, sampling schemes, inclusion probabilities and estimation. Review of important results in simple and stratified random sampling, Fixed (Design –based) and Superpopulation (modelbased) approaches.

Unit II: (11 Hours)

Sampling with varying probabilities (unequal probability sampling) with or without replacement: pps, πps and non- πps sampling procedures and estimation based on them, Nonnegative variance estimation.

Unit III: (12 Hours)

Two-way stratification, post-stratification, controlled sampling, Estimation based on auxiliary data (involving one or more auxiliary variables) under design-based and model-based approaches, Double (two-phase) sampling with special reference to the selection with unequal probabilities in at least one of the phases.

Unit IV: (12 Hours)

Systematic sampling and its application to structured populations, Cluster sampling (with varying sizes of clusters), Two-stage sampling (with varying sizes of first-stage units), Warner's and Simmons' randomized response techniques for one qualitative sensitive characteristic.

Suggested Readings:

1. Cochran, W.G. (2011). Sampling Techniques, 3rd Ed., Wiley Eastern John Wiley and Sons.
2. Murthy M.N. (1977). Sampling Theory & Statistical Methods, Statistical Pub. Society, Calcutta.

3. Singh, D. and Chaudhary, F. S. (2015). Theory and Analysis of Sample Survey Designs.
4. Mukhopadhyay, P. (2009). Theory and Methods of Survey Sampling, 2nd Edn., Prentice Hall of India, New Delhi.

Teaching Plan:

Week 1-2:	Basic ideas and distinctive features of sampling, Probability sampling designs, sampling schemes, inclusion probabilities and estimation.
Week 3:	Review of important results in simple and stratified random sampling.
Week 4-5:	Fixed (Design –based) and Superpopulation (modelbased) approaches.
Week 6-7:	Sampling with varying probabilities (unequal probability sampling) with or without replacement: pps, πps and non- πps sampling procedures and estimation based on them
Week 8:	Nonnegative variance estimation.
Week 9:	Two-way stratification, post-stratification, controlled sampling
Week 10:	Estimation based on auxiliary data (involving one or more auxiliary variables) under design-based and model-based approaches.
Week 11:	Double (two-phase) sampling with special reference to the selection with unequal probabilities in at least one of the phases.
Week 12:	Systematic sampling and its application to structured populations, Cluster sampling (with varying sizes of clusters).
Week 13:	Two-stage sampling (with varying sizes of first-stage units),
Week 14:	Randomized response techniques

List of Practicals:

1. To select SRS with and without replacement.
2. For SRSWOR and SRSWR, estimate mean, standard error, the sample size.
3. Ratio and Regression estimation: Calculate the population mean or total of the population. Calculate mean squares. Compare the efficiencies of ratio and regression estimators relative to SRS.
4. Cluster sampling: estimation of mean or total, variance of the estimate, estimate of intra-class correlation coefficient, efficiency as compared to SRS.

5. PPS (Probability Proportional to Size) and πps (Horvitz-Thompson framework)

Discipline -Specific Elective (DSE) Courses

Discipline -Specific Elective (DSE) Courses- 1a: Analysis

Course Title and Code	Credits	Credit Distribution of Course		
		Lecture (45 Hours)	Tutorial (15 Hours)	Practical
DSE 1a: Analysis	4	3	1	0

Course Objectives: The main objective of this course is to introduce students the knowledge of real field and complex field with their properties and relativity between complex plane and real line. These properties and relations provide grounds for Probability Theory and help in theoretical research in Statistics.

Course Learning Outcomes:

After successful completion of this course, student will be able to:

1. Understand existence of integral and their evaluation.
2. Apply convergence theorems of sequence and series of real valued function and complex valued functions.
3. Understand change of multiple integral into line integral.
4. Learn how apply real and complex-analytic methods to problems in probability theory.
5. Understand complex region and relativity between complex plane and real line.
6. Analyze power series, Laurent series, and residue calculus.
7. Solve contour integrals.
8. Gain exposure to challenging exercises that deepen theoretical understanding.

Unit I: (10 Hours)

Functions of bounded variation, Riemann integration and Riemann-Stieltjes integration, Statement of the standard property and problem based on them, Multiple integrals, repeated integrals, Change of variables in multiple integration.

Unit II: (11 Hours)

Differentiation under integral sign, Leibnitz rule, Dirichlet integral, Liouville's extension, Uniform convergence of sequence of functions and series of functions, Cauchy's criteria and Weirstrass M-test, Maxima-minima of functions of several variables.

Unit III: **(12 Hours)**

Properties of complex numbers, Region in complex plane, Limit, continuity and differentiability of function of complex variables, Analytic function, Contour integration, Cauchy integral formula, Liouville's theorem, Fundamental theorem of Algebra.

Unit IV: (12 Hours)

Power series and radius of convergence, Taylor's and Laurent's series, Singular points and their types, Residue at singular point and residue at infinity, Cauchy residue theorem, Evaluation of real integrals involving sine and cosine using residue.

Suggested Readings:

1. Brown, J. W. and Chirchill, R. V. (2009). Complex variables and Applications, McGraw Hill.
2. Bartle, R. G. (1976). Elements of Analysis, John Wiley & Sons.
3. Bak, J. and Newman, D. J. (1997). Complex Analysis, Springer.
4. Rudin, W. (1985). Principles of Mathematical Analysis, McGraw Hill.
5. Rose, K. A. (2004). Elementary Analysis: The Theory of Calculus, Springer (SIE).

Teaching Plan:

Week 1:	Functions of bounded variations. Total variation. Positive variation and negative variation. Expression of a function of bounded variation in terms of monotonically increasing functions.
Week 2-3:	Riemann integration. Inequalities of upper and lower sums. Riemann conditions of integrability. Riemann sum and definition of Riemann integral through Riemann sums. Riemann integrability of continuous functions. Monotonic functions and functions with finite number of discontinuities.
Week 4:	Intermediate value theorem for integrals. Fundamental theorem of Calculus. Riemann-Stieltjes integral. Evaluating Riemann-Stieltjes integral using definition and also, by reducing it to Riemann integral.

Week 5:	Multiple integrals and their evaluation by repeated integration. Change of variable in multiple integration. Differentiation under integral sign. Leibnitz rule. Dirichlet integral. Liouville's extension.
Week 6:	Pointwise and uniform convergence of sequence of functions. Cauchy's criteria and Weirstrass M-test. Continuity, differentiability and integrability of a limit function.
Week 7:	Uniform convergence of series of functions. Conditions for continuity, differentiability and integrability of the sum function. Maxima-minima of functions of several variables.
Week 8-9:	Properties of complex numbers. Region in the complex plane. Functions of complex variable. Limit, continuity and differentiability of functions of complex variable. Cauchy Riemann equations. Sufficient conditions for differentiability. Analytic functions and examples of analytic functions.
Week 10:	Contour integrals and its example. Upper bounds for moduli of contour integrals. M-L formula. Antiderivatives. Cauchy theorem. Cauchy-Goursat theorem and Cauchy integral formula.
Week 11:	Liouville's theorem and fundamental theorem of Algebra. Convergence of sequence and series. Absolute and uniform convergence of power series.
Week 12:	Taylor's series and Laurent's series and their examples. Uniqueness of series representation of power series.
Week 13:	Singular points. Classification of singularity. Residue at poles and its examples. Residue at infinity with examples.
Week 14:	Cauchy residue theorem. Evaluation of definite integrals and real integrals involving sine and cosines.

Discipline-Specific Elective (DSE) Courses - 1b: Time Series Analysis

Course Title & Code	**Credits**	**Credit Distribution of the Course**		
		Lecture (45 Hours)	**Tutorial**	**Practical/ Practice (30 Hours)**

DSE 1b: Time Series Analysis	4	3	0	1

Course Objectives: The main objective is to teach the time series modelling and the concept of forecasting and future planning.

Course Learning Outcomes: After completing the paper, students will be able to apply:

1. Time series analysis concepts to practical data scenarios.
2. Techniques to identify and analyze trends and seasonality.
3. Various time series models, including MA, AR, ARMA, and ARIMA, for data modeling.
4. Time series models for effective forecasting.
5. Information criteria (AIC, BIC) to select the most suitable models.
6. Yule-Walker equations to analyze AR processes.
7. Methods to address nonstationarity in time series data.
8. The random walk model and conduct the Dickey-Fuller test for unit root analysis.

Unit I: (10 Hours)

Time series as a stationary or nonstationary stochastic process, sample autocovariance function (acvf) and autocorrelation function (acf) at lag k, partial autocorrelation function (pacf), correlogram, lag operators and linear filters, Ergodicity and Stationarity.

Unit II: (12 Hours)

Wald decomposition, Generic linear process and its acvf, acf. Autoregressive (AR) process, moving average (MA) process, acf and pacf for AR and MA processes, Yule-Walker equations for AR processes.

Unit III: (12 Hours)

Stationarity and invertibility conditions, ARIMA (p,d,q) model, estimation of parameters for AR, MA, ARMA and ARIMA processes, identification of processes with ACF PACF, Model estimation techniques-AIC, AICC, BIC, etc.

Unit IV: (11 Hours)

Forms of nonstationarity in time series, random walk model, Dickey-Fuller test for unit root. ARCH and GARCH Processes, order identification, estimation, diagnostic.

Suggested Readings:

1. Box, George E. P., Gwilym M. Jenkins, Gregory C. Reinsel, Greta M. Ljung . (2015) Time Series Analysis – Forecasting and Control, 5th Edition, Wiley.
2. Brockwell, P. J. and Davies, R. A. (2009) Introduction to Time Series and Forecasting (2nd Edition Indian Print). Springer.
3. Chatfield, C. (1975) The Analysis of Time series: Theory and Practice. Fifth Ed. Chapman and Hall.
4. Chatfield, C. (2003) Analysis of Time Series, An Introduction, 6th Edition, CRC Press.
5. Jonathan, D. C. and Kung, S.C. (2008). Series Analysis with application in R. Second Ed. Springer
6. Kendall, M. G. and Ord, J. K. Time Series (Third edition), Edward Arnold.
7. Montgomery, D. C. and Johnson, L. A. (1977). Forecasting and Time series Analysis, McGraw Hill.
8. Montgomery, D.C., Jennings, C. and Kulahci, M. (2016). Introduction to Time Series Analysis and Forecasting. Second Ed., Wiley.
9. Shumway, R.H. and Stoffer, D.S. (2017). Time Series Analysis and Its Applications: With R Examples. Fourth Edition. Springer.

Teaching Plan:

Week 1:	Time series as a stationary or nonstationary stochastic process, sample autocovariance function (acvf)
Week 2:	autocorrelation function (acf) at lag k, partial autocorrelation function (pacf)
Week 3:	correlogram, lag operators and linear filters, Ergodicity and Stationarity
Week 4:	Wald decomposition, Generic linear process and its acvf, acf
Week 5:	Autoregressive (AR) process, moving average (MA) process
Week 6:	acf and pacf for AR and MA processes,
Week 7:	Yule-Walker equations for AR processes.
Week 8:	Stationarity and invertibility conditions, ARIMA (p,d,q) model

Week 9:	Estimation of parameters for AR, MA, ARMA and ARIMA processes
Week 10:	Identification of processes with ACF PACF,
Week 11:	Model estimation techniques-AIC, AICC, BIC, etc.
Week 12:	Forms of nonstationarity in time series, random walk model
Week 13:	Dickey-Fuller test for unit root. ARCH and GARCH Processes
Week 14:	Order identification, estimation, diagnostic tests.

List of Practicals:

1. Calculate and plot descriptive statistics (mean, variance, autocorrelation, partial autocorrelation), create time plots.
2. Identify potential trends and seasonality.
3. Identify potential ARIMA(p,d,q) models based on ACF/PACF plots.
4. Identify potential ARIMA(p,d,q) models based on unit root tests.
5. Compare the performance of different ARIMA models (e.g., different orders) using information criteria (AIC, BIC).
6. Implementing Dickey-Fuller (or other unit root) tests on various datasets to determine stationarity.

Course Title & Code	**Credits**	**Discipline-Specific Elective (DSE) Course- 1c: Bio-Statistics** **Credit Distribution of the Course**		
		Lecture (45 Hours)	**Tutorial**	**Practical/ Practice (30 Hours)**
DSE 1c: Bio-Statistics	**4**	**3**	**0**	**1**

Course Objectives: Biostatistics is one area of Applied Statistics that is concerned with the application of statistical methods to medical, biological, epidemiological and health related aspects of all forms of life.

Course Learning Outcomes: After successful completion of this course, student will be able to:

1. Develop understanding of time-to-event data in Biomedical Sciences.
2. Summarizing clinical data using displays, parametric and non- parametric approaches.
3. Understanding concepts of conditional and inverse probabilities as applied to survival data.
4. Establishing meaningful relationships for causative and consequential health factors.
5. Understand survival patterns in presence and in absence of censoring.
6. Account for censored patterns and their implications.
7. Estimation of failure and hazard forms based on patient data records.
8. Formulate and interpret stochastic models for specific-disease data sets.
9. Comprehend basis and construction of clinical trials for different stages.
10. Analyze concept of Biometric genetics.

Unit I: (11 Hours)

Analysis of Medical, Epidemiologic and Clinical Data: Studying association between a disease and a characteristic: (a) Types of studies in Epidemiology and Clinical Research (i) Prospective study (ii) Retrospective study (iii) Cross-sectional data, (b) Dichotomous Response and Dichotomous Risk Factor: 2x2 Tables (c) Expressing relationship between a risk factor and a disease (d) Inference for relative risk and odds ratio for 2x2 table, Sensitivity, specificity and predictivities. Clinical Trials: Its Planning and its Four Phases.

Unit: (11 Hours)

Special Survival Features: Censoring and its types. Study time and Patient time. Survival Analysis: Survival Distributions and their Properties *viz.* Exponential, Weibull, Gamma, Rayleigh and Lognormal. Estimation of Survivor and Hazard Functions: Life Table, Kaplan-Meier and Nelson-Aalen Estimates. Estimating Median and Survival Times. Estimation of Mean survival time and variance for Type I and Type II Censored data with examples.

Unit III: (11 Hours)

Cox-Proportional Hazards Model: Its Linear Component, Fitting, Hypothesis Tests. Estimating Hazard and Survivor Function. Kaplan Meier Estimate, Hazard and Cumulative Incidence Functions, Modelling. Cause Specific Hazard and Incidence, Model Checking. Sample Size Requirements for a Survival Study.

Unit IV: (12 Hours)

Multiple Factor Hypothesis for Process of Heredity. Medelian Population: Gene Frequency and Genotype Frequency. Hardy Weinberg Law: Multiple Alleles, Two or More Pairs of Genes, Linkage of Genes, Sex Linked Genes. Influence of Gene Frequencies on Population Mean. Breeding Value of Genotypes. Dominance Deviation.

Suggested Readings:

1. Biswas, S. (1995). Applied Stochastic Processes: A Biostatistical and Population Oriented Approach, Wiley Eastern Ltd.
2. Collett, D. (2003). Modelling Survival Data in Medical Research, Chapman & Hall/CRC.
3. Cox, D.R. and Oakes, D. (1984). Analysis of Survival Data, Chapman and Hall.
4. Dabholkar A.R. (1999) Elements of Bio Metrical Genetics. Concept Publishing Co. , New Delhi.
5. Elandt Johnson R.C. (1971). Probability Models and Statistical Methods in Genetics, John Wiley & Sons.
6. Ewens, W. J. (1979). Mathematics of Population Genetics, Springer Verlag.
7. Ewens, W. J. and Grant, G.R. (2001). Statistical methods in Bio informatics: An Introduction, Springer.
8. Friedman, L.M., Furburg, C. and DeMets, D.L. (1998). Fundamentals of Clinical Trials, Springer Verlag.
9. Gross, A. J. And Clark V.A. (1975). Survival Distribution; Reliability Applications in Biomedical Sciences, John Wiley & Sons.
10. Indrayan, A. (2008). Medical Biostatistics, 2nd ed., Chapman & Hall/CRC.
11. Lee, Elisa, T. (1992). Statistical Methods for Survival Data Analysis, John Wiley & Sons.
12. Li, C.C. (1976). First Course of Population Genetics, Boxwood Press.
13. Liu Xian. (2012) Survival Analysis: Model and Applications. Wiley.
14. Miller, R.G. (1981). Survival Analysis, John Wiley & Sons.
15. Robert F. Woolson (1987). Statistical Methods for the analysis of biomedical data, John Wiley & Sons.
16. Tattar P.N and Vaman H.J. (2023) Survival Analysis. CRC Press.

Teaching Plan:

Week 1:	Analysis of Medical, Epidemiologic and Clinical Data: Studying association between a disease and a characteristic: (a) Types of studies in Epidemiology and Clinical Research
Week 2:	Prospective study (ii) Retrospective study (iii) Cross-sectional data, (b) Dichotomous Response and Dichotomous Risk Factor: 2x2 Tables
Week 3:	Expressing relationship between a risk factor and a disease (d) Inference for relative risk and odds ratio for 2x2 table
Week 4:	Sensitivity, specificity and predictivities. Clinical Trials: Its Planning and its Four Phases.
Week 5:	Special Survival Features: Censoring and its types. Study time and Patient time.
Week 6:	Survival Analysis: Survival Distributions and their Properties *viz.* Exponential, Weibull, Gamma, Rayleigh and Lognormal.
Week 7:	Estimation of Survivor and Hazard Functions: Life Table, Kaplan-Meier and Nelson-Aalen Estimates.
Week 8:	Estimation of Mean survival time and variance for Type I and Type II Censored data with examples.
Week 9:	Cox-Proportional Hazards Model: Its Linear Component, Fitting, Hypothesis Tests. Estimating Hazard and Survivor Function.
Week 10:	Kaplan Meier Estimate, Hazard and Cumulative Incidence Functions, Modelling.
Week 11:	Cause Specific Hazard and Incidence, Model Checking. Sample Size Requirements for a Survival Study.
Week 12:	Multiple Factor Hypothesis for Process of Heredity. Medelian Population: Gene Frequency and Genotype Frequency.
Week 13:	Hardy Weinberg Law: Multiple Alleles, Two or More Pairs of Genes, Linkage of Genes, Sex Linked Genes.
Week 14:	Influence of Gene Frequencies on Population Mean. Breeding Value of Genotypes. Dominance Deviation.

List of Practicals:

1. Interpreting clinical Trial data.
2. Sample size estimation in clinical Trials.

3. Plotting Survival and Hazard Curves for different parameter combinations in respect of some life time distributions.
4. Computing Kaplan-Meier estimates based on recorded surviving times with and without censoring.
5. Fitting of Cox-Proportional Hazard Model.
6. Hypothesis Formulation and their Testing for Cox-Proportional Hazard Model.
7. Estimation of Mean Survival Time and its variance for complete and survival data.
8. Random union among gametes.
9. Gene effect on population mean.

Discipline-Specific Elective (DSE) Courses-
1d: Official And National Development Statistics

Course Title & Code	Credits	Credit Distribution of the Course		
		Lecture (45 Hours)	Tutorial (15 Hours)	Practical/ Practice
DSE 1d: Official and National Development Statistics	**4**	**3**	**1**	**0**

Course Objectives: This course will provide the important information on Indian Official Statistical System.

Course Learning Outcomes:

After successful completion of this course, student will be able to:

1. Learn about the role and function of National and State Statistical Organizations.
2. Knowing important sectors of Indian official statistics system (National and State) with their important regular publications.
3. Understanding important data collection mechanism in different sectors.
4. Finding important reasons for non-response while collecting Official Statistics.
5. Learning concepts of National Accounts with the release.
6. Finding statistics related to industries, foreign trade, balance of payment, cost of living,

inflation, educational and other social statistics.

7. Knowing socio-economic indicators, gender awareness/statistics, important surveys, and censuses pertaining to official statistics.

UNIT I: (11 Hours)

An overview of national and international statistical systems. National Statistical Organization: Vision and Mission, Central Statistical Office (CSO), National Sample Survey Office (NSSO); roles and responsibilities; important publications. Indian State statistical organizations: Important role, function and activities. Organization of large scale sample surveys.

UNIT II: (12 Hours)

National Statistical Commission: Need, constitution, its role, functions, etc.; Legal acts/ provisions/support for Official Statistics.. Data collection & compilation mechanism, processing, analysis and dissemination systems, agencies involved. Population growth in developed and developing countries, evaluation and performance of family welfare programmes.

UNIT III: (10 Hours)

Scope and content of population census of India, method of data collection. Sector wise statistics: Agriculture, Environment and Forestry, Health, Education, Women, and Child, etc. Important surveys & censuses, indicators, agencies, and usages, etc.

UNIT IV: (12 Hours)

National Accounts: Definition, basic concepts, issues, the strategy, collection of data and release. System of collection of agricultural and forestry statistics, crop forecasting and estimation, productivity, fragmentation of holdings, support prices. Statistics related to industries, foreign trade, balance of payment, cost of living, inflation, educational and other social statistics. Socio-Economic Indicators, Gender Awareness/Statistics.

Suggested Readings:

1. Saluja, M.R. (2017). Measuring India. The Nation's Statistics System, OUP Catalogue, Oxford University Press.
2. Directorate General of Commercial Intelligence and Statistics (DGCIS). Monthly Statistics of the Foreign Trade of India. Calcutta: DGCIS.
3. Panse, V.G. (1954): Estimation of Crop Yields. Food and Agricultural Organization.
4. UNESCO. Principles and recommendations for population and housing censuses. Revision 3. New York: United Nations.

5. Ministry of Statistics and Programme Implementation (MoSPI). Statistical System in India. Government of India.
6. Reserve Bank of India. Handbook of Statistics on the Indian Economy. Mumbai: Reserve Bank of India
7. Ministry of Agriculture and Farmer's Welfare, Government of India. Pocket Book of Agricultural Statistics.

Forest Survey of India (2023). India State of Forest Report. Dehradun: Forest Survey of India

Teaching Plan:

Week 1-2:	An overview of National and International statistical systems. National Statistical Organization: Vision and Mission, Central Statistical Office (CSO), National Sample Survey Office (NSSO).
Week 3:	Roles and responsibilities of CSO and NSSO; Important publications. State Indian statistical organizations
Week 4:	Important role, function and activities of Indian State statistical organizations. Organization of large scale sample surveys.
Week 5:	National Statistical Commission: Need, constitution, its role, functions, etc.
Week 6:	Legal acts/ provisions/ support for Official Statistics.
Week 7:	Data collection & compilation mechanism, processing, analysis and dissemination systems, agencies involved,
Week 8:	Population growth in developed and developing countries, evaluation and performance of family welfare programmes.
Week 9:	Scope and content of population census of India, method of data collection. Sector wise statistics: Agriculture, Environment and Forestry, Health, Education, Women, and Child, etc.
Week 10:	Important surveys & censuses, indicators, agencies, and usages, etc.
Week 11:	National Accounts: Definition, basic concepts, issues, the strategy, collection of data and release.
Week 12:	System of collection of agricultural and forestry statistics, crop forecasting and estimation, productivity, fragmentation of holdings, support prices.

Week 13:	Statistics related to industries, foreign trade, balance of payment, cost of living, inflation, educational and other social statistics.
Week 14:	Socio-Economic indicators, Gender Awareness/Statistics.

Generic Elective (GE)- 1a: Statistical Computing Using R

Course Title & Code	Credits	Credit Distribution of the Course		
		Lecture (45 Hours)	Tutorial	Practical/ Practice (30 Hours)
GE 1b: Statistical Computing Using R	**4**	**3**	**0**	**1**

Course Objectives: The main objectives of this course are:

1. To learn the principles and methods of data analysis.
2. To provide a basic understanding of methods of analysing data from different fields.
3. To perform data analysis using R software.

Course Learning Outcomes: After successful completion of this course, the students will be able to:

1. Carry out data analysis using R software.
2. Effectively visualize and summarize the data.
3. Interpret the results of statistical analysis.

UNIT I: (11 Hours)

Introduction to R: Installing R, R console, Script file, Workspace, Getting help, R packages, Installing and loading packages. R data structures: vectors, matrices, array, data frames, factors, lists. Creating datasets in R, Importing and exporting dataset, annotating datasets. Graphs**:** Creating and saving graphs, customizing symbols, lines, colors and axes, combining multiple graphs into one, bar plots, boxplot and dot plots, pie chart, stem and leaf display, histogram and kernel density plots. Data management**:** Manipulating dates and missing values, understanding data type conversion, creating and recoding variables, sorting, merging and sub-setting data sets. Mathematical and statistical functions, character functions, looping and conditional statements, user defined functions.

UNIT II: (11 Hours)

Basic statistics: Descriptive statistics, frequency and contingency tables, outlier detection, testing of normality, basics of statistical inference in order to understand hypothesis testing, p-value and confidence intervals. Parametric tests: Tests for population mean and variance for two or more populations, tests for independence and measures of association, sample size determination for common statistical methods using pwr package. Nonparametric tests.

UNIT III: (11 Hours)

Correlation: Correlations between quantitative variables and their associated significance tests. Regression Analysis: Fitting simple and multiple regression model forward, backward and stepwise regression, polynomial regression, regression diagnostics to assess the statistical assumptions, methods for modifying the data to meet these assumptions more closely, selecting a final regression model from many competing models. ANOVA: Fitting and interpreting ANOVA type models, evaluating model assumptions, basic experimental designs: CRD, RBD, LSD and factorial experimental designs.

UNIT IV: (12 Hours)

Time series Analysis: Creating and manipulating a time series, Components of a time series, auto-correlation and partial correlation function, estimating and eliminating the deterministic components of a time series. Developing Predictive Models: Forecasting using exponential models, predictive accuracy measures for time-series forecast, testing for stationarity, Forecasting using ARMA and ARIMA models. EM algorithm: Applications to missing and incomplete data problems, mixture models.

Suggested Readings:

1. Crawley, M.J. (2013). The R Book, 2nd ed., John Wiley.
2. Davies, T. M. (2016). The Book of R: A First Course in Programming and Statistics, No Starch Press, San Francisco.
3. Field, A., Miles, J. and Field, Z. (2012). Discovering Statistics using R, Sage, Los Angels.
4. Kabacoff, R.I. (2015). R in Action: Data Analysis and Graphics in R, 2nd ed., Manning

Publications.

Teaching Plan:

Week 1-2:	Introduction to R: Installing R, R console, Script file, Workspace, Getting help, R packages, Installing and loading packages. R data structures: vectors, matrices, array, data frames, factors, lists. Creating datasets in R, Importing and exporting dataset, annotating datasets. Graphs: Creating and saving graphs, customizing symbols, lines, colors and axes, combining multiple graphs into one, bar plots, boxplot and dot plots, pie chart, stem and leaf display, histogram and kernel density plots.
Week 3-4:	Data management: Manipulating dates and missing values, understanding data type conversion, creating and recoding variables, sorting, merging and sub- setting data sets. Mathematical and statistical functions, character functions, looping and conditional statements, user defined functions.
Week 5-6:	Basic statistics: Descriptive statistics, frequency and contingency tables, outlier detection, testing of normality, basics of statistical inference in order to understand hypothesis testing, computing p-value and confidence intervals.
Week 7-8:	Parametric tests: Tests for population mean and variance for two or more populations, tests for independence and measures of association, sample size determination for common statistical methods using pwr package. Nonparametric tests.
Week 9-10:	Correlation: Correlations between quantitative variables and their associated significance tests. Regression Analysis: Fitting simple and multiple regression models, forward, backward and stepwise regression, polynomial regression, regression diagnostics to assess the statistical assumptions, methods for modifying the data to meet these assumptions more closely, selecting a final regression model from many competing models.
Week 11:	ANOVA: Fitting and interpreting ANOVA type models, evaluating model assumptions, basic experimental designs: CRD, RBD, LSD and factorial experimental designs.

Week 12-14:	Creating and manipulating a time series, Components of a time series, auto-correlation and partial correlation function, estimating and eliminating the deterministic components of a time series. Developing Predictive Models: Forecasting using exponential models, predictive accuracy measures for time-series forecast, testing for stationarity, Forecasting using ARMA and ARIMA models. EM algorithm: For missing and incomplete data problems.

List of Practicals:

1. Problems based on creating vectors and mathematical operations.
2. Problems based on sequences, replications, sorting and lengths.
3. To perform matrix operations, importing and exporting datasets.
4. Basic plotting of R graphical functionality.
5. Basic statistics and testing of hypothesis.
6. Parametric and non-parametric tests.
7. Problems based on sample size determination.
8. Correlation and regression analysis using quantitative variables.
9. Analysis of variance and basic design of experiments (CRD, RBD, LSD and factorial designs).
10. To plot a time series function, autocorrelation function and correlogram.
11. Problems based on ARMA and ARIMA models.
12. Problems based on EM algorithms for missing and incomplete data problems.

Skill Based/Specialized Laboratory Courses

Skill Based/Specialized Laboratory (SB) Courses- 1a: Data Analysis Using Excel

Course Title & Code	Credits	Credit Distribution of the Course		
		Lecture	Tutorial	Practical/ Practice (60 Hours)
SB 1a: Data Analysis Using Excel	2	0	0	2

Course Objectives

This course is designed to provide students with a solid understanding of fundamental statistical concepts and practical experience in using Microsoft Excel for data analysis. By the end of the course, students will be proficient in applying statistical techniques to interpret data and make decisions using Excel tools.

Course Learning Outcomes

After successful completion of this course, student will be able to:

1. Grasp fundamental statistical concepts and their real-world applications.
2. Conduct both descriptive and inferential statistical analyses using Excel functions and tools.
3. Analyze, interpret, and effectively present statistical findings.

Unit 1: (15 Hours)

Introduction to MS Excel: Interface, functions, and statistical functions, data analysis ToolPak for statistical analysis, descriptive statistics, basic matrix operations, addition, multiplication, transpose, determinant, inverse, eigenvalues and eigenvectors, problem solving using Excel functions for sampling techniques, including simple random sampling, stratified random sampling, systematic sampling, ratio and regression estimation, cluster sampling and two-stage sampling methods.

Unit 2: (15 Hours)

Hypothesis testing using Excel- z-test for single mean, difference of two means and related confidence intervals, t-test for single mean, difference of two means, paired t-test and related confidence intervals, t-test for correlation coefficient.

Unit 3: (15 Hours)

Hypothesis testing using Excel- chi-square test for single variance, chi-square test for independence of attributes, chi-square test for testing goodness of fit, Bartlett's test, F-test for difference of two variances and related confidence intervals.

Unit 4: (15 Hours)

Advanced problem-solving using Excel functions and the data analysis ToolPak for One and two-Way ANOVA, completely randomized design (CRD), randomized block design (RBD), Latin square design (LSD) and analysis of RBD and LSD with missing observations.

Suggested readings:

1. Kanji, G. K. (2006). 100 Statistical Tests. SAGE Publications, London.
2. Montgomery, D. C. (2013). Design and Analysis of Experiments. John Wiley & Sons, New York.
3. Panneerselvam, R. (2024). Business Statistics Using Excel: A Complete Course in Data Analytics. Routledge, New York.
4. Rajagopalan, V. (2006). Selected Statistical Tests. New Age International Publishers, New Delhi.
5. Schmuller, J. (2009). Statistical Analysis with Excel for Dummies. Wiley, Indiana.
6. Searle, S. R., & Khuri, A. I. (2017). Matrix Algebra Useful for Statistics. John Wiley & Sons, New York.

SEMESTER II

Discipline-Specific Core (DSC) Course- 2a: Statistical Inference

Course Title & Code	Credits	Credit Distribution of the Course		
		Lecture (45 Hours)	Tutorial (15 Hours)	Practical/ Practice
DSC 2a: Statistical Inference	**4**	**3**	**1**	**0**

Course Objectives: To make students aware of estimation (both point and interval) and testing (both simple and composite hypotheses) procedures.

Course Learning Outcomes:

After successful completion of this course, student will be able to:

1. Apply various estimation and hypothesis testing procedures to deal with real life problems.
2. Demonstrate a comprehensive understanding of Fisher Information, Lower bounds to variance of estimators, MVUE.
3. Explain and apply the Neyman-Pearson fundamental lemma, develop UMP tests, and perform interval estimation, including the construction of confidence intervals.

Unit I: (12 Hours)

Minimal sufficiency and ancillarity, Exponential families and Pitman families, Invariance property of Sufficiency under one-one transformations of sample and parameter spaces. Fisher Information for one and several parameters models. Lower bounds to variance of estimators for one and several parameters models, necessary and sufficient conditions for MVUE.

Unit II: (12 Hours)

Neyman-Pearson fundamental lemma and its applications, UMP tests for simple null hypothesis against one-sided alternatives and for one-sided null against one-sided alternatives in one parameter exponential family. Extension of these results to Pitman family when only upper or lower end depends on the parameters and to distributions with MLR property.

Unit III: (10 Hours)

Non-existence of UMP tests for simple null against two-sided alternatives in one parameter exponential family. Families of distributions with monotone likelihood ratio and UMP tests.

Unit IV: (11 Hours)

Interval estimation, confidence level, construction of shortest expected length confidence interval, uniformly most accurate one-sided confidence Interval and its relation to UMP tests for one-sided null against one-sided alternative hypotheses.

Suggested Readings:

1. Bartoszynski, R. and Bugaj, M.N. (2007). Probability and Statistical Inference, John Wiley & Sons.
2. Ferguson, T.S. (1967). Mathematical Statistics, Academic Press.
3. Kale, B.K. (1999). A First Course on Parametric Inference, Narosa Publishing House.
4. Lehmann, E.L. (1986). Theory of Point Estimation, John Wiley & Sons.
5. Lehmann, E.L. (1986). Testing Statistical Hypotheses, John Wiley & Sons.
6. Rohatgi, V.K. and Saleh, A.K. Md. E. (2005). An Introduction to Probability and Statistics, 2nd Edn., John Wiley & Sons.
7. Rao, C.R. (1973). Linear Statistical Inference and Its Applications, Wiley Eastern Ltd., New Delhi.
8. Zacks, S. (1971). Theory of Statistical Inference, John Wiley & Sons.

Teaching Plan:

Week 1-2:	Minimal sufficiency and ancillarity.
Week 3:	Exponential families and Pitman families.
Week 4:	Invariance property of Sufficiency under one-one transformations of sample and parameter spaces.
Week 5:	Fisher Information for one and several parameters models.
Week 6:	Lower bounds to variance of estimators for one and several parameters models, necessary and sufficient conditions for MVUE.
Week 7:	Neyman-Pearson fundamental lemma and its applications.
Week 8:	UMP tests for simple null hypothesis against one-sided alternatives and for one-sided null against one-sided alternatives in one parameter exponential family.
Week 9:	Extension of these results to Pitman family when only upper or lower end depends on the parameters and to distributions with MLR property.
Week 10:	Non-existence of UMP tests for simple null against two-sided alternatives in one parameter exponential family.
Week 11:	Families of distributions with monotone likelihood ratio and UMP tests.
Week 12:	Interval estimation, confidence level, construction of shortest expected length confidence interval.

Week 13:	Uniformly most accurate one-sided confidence Interval.
Week 14:	Its relation to UMP tests for one-sided null against one-sided alternative hypotheses.

Discipline-Specific Core (DSC) Course- 2b: Design of Experiments

Course Title & Code	Credits	Credit Distribution of the Course		
		Lecture (45 Hours)	Tutorial	Practical/Practice (30 Hours)
DSC 2b: Design of Experiments	4	3	0	1

Course Objectives: This course provides the students the ability to formulate the design and conduct experiments, as well as to analyze and interpret data.

Course Learning Outcomes: After successful completion of this course, student will be able to:

1. Apply ANOVA for two –way classification, fixed effect models with equal, unequal and proportional number of observations per cell, Random and Mixed effect models with m (>1) observations per cell.
2. Design and analyse incomplete block designs, understand the concepts of orthogonality, connectedness and balance.
3. Understand the concepts of finite fields and finite geometries and apply them in construction of MOLS, construction of balanced incomplete block designs, confounded factorial experiments.
4. Identify the effects of different factors and their interactions and analyse factorial experiments.
5. Construct complete and partially confounded factorial designs and perform their analysis.

6. Apply Split-plot designs and their analysis in practical situations.
7. Understand the effects of independence or dependence of different factor under study.

Unit I: (12 Hours)

Review of linear estimation and basic designs. Elimination of heterogeneity in two directions. ANOVA: Fixed effect models (2-way classification with equal, unequal and proportional number of observations per cell), Random and Mixed effect models (2-way classification with m (>1) observations per cell).

Unit II: (12 Hours)

Incomplete Block Designs. Concepts of Connectedness, Orthogonality and Balance. Intrablock analysis of Generic Incomplete Block design. B.I.B designs with and without recovery of interblock information.

Unit III: (11 Hours)

Finite fields. Finite Geometries- Projective geometry and Euclidean geometry. Construction of complete set of mutually orthogonal latin squares. Construction of B.I.B.D. using finite Abelian groups, MOLS, finite geometry and method of differences.

Unit IV: (10 Hours)

Symmetrical factorial experiments (sm, where s is a prime or a prime power), Confounding in sm factorial experiments through pencils, sk-p fractional factorial where s is a prime or a prime power. Split-plot experiments.

Suggested Readings:

1. Chakrabarti, M.C. (1962). Mathematics of Design and Analysis of Experiments, Asia Publishing House, Bombay.
2. Das, M.N. and Giri, N.C. (1986). Design and Analysis of Experiments, Wiley Eastern Limited.
3. Dean, A. and Voss, D. (1999). Design and Analysis of Experiments, Springer. First Indian Reprint 2006.
4. Dey, A. (1986). Theory of Block Designs, John Wiley & Sons.
5. Hinkelmann, K. and Kempthorne, O. (2005). Design and Analysis of Experiments, Vol.

2: Advanced Experimental Design, John Wiley & Sons.

6. John, P.W.M. (1971). Statistical Design and Analysis of Experiments, Macmillan Co., New York.
7. Kshirsagar, A.M. (1983). A Course in Linear Models, Marcel Dekker, Inc., N.Y.
8. Montgomery, D. C. (2005). Design and Analysis of Experiments, 6th ed., John Wiley & Sons.
9. Raghavarao, D. (1970). Construction and Combinatorial Problems in Design of Experiments, John Wiley & Sons.
10. Raghavarao, D. and Padgett, L. V. (2005). Block Designs: Analysis, Combinatorics, and Applications, World Scientific.

Teaching Plan:

Week 1:	Review of linear estimation and basic designs, elimination of heterogeneity in two directions.
Week 2:	ANOVA, Fixed effect models (2-way classification with equal, unequal and proportional number of observations per cell).
Week 3:	ANOVA, Random and Mixed effect models (2-way classification with m (>1) observations per cell).
Week 4-5:	Incomplete Block Designs. Concepts of Connectedness, Orthogonality and Balancedness.
Week 6-7:	Intrablock analysis of Generic Incomplete Block design. B.I.B designs with and without recovery of inter block information.
Week 8:	Finite fields.
Week 9:	Finite Geometries- Projective geometry and Euclidean geometry.
Week 10:	Construction of complete set of mutually orthogonal Latin squares. Construction of B.I.B.D. using finite Abelian groups and MOLS.
Week 11:	Construction of B.I.B.D using finite geometry and method of differences.
Week 12-13:	Symmetrical factorial experiments (s^m, where s is a prime or a prime power), Confounding in s^m factorial experiments through Pencils.
Week 14:	S^{k-p} fractional factorial where s is a prime or a prime power. Split-plot experiments.

List of Practicals:

1. ANOVA for two –way classification:
   i. Fixed effect model: equal, unequal and proportional number of observations.
   ii. Random effect model with 'm' observations per cell.
   iii. Mixed effect model with 'm' observations per cell.
2. IBD:
   i. Intrablock analysis
   ii. Inter block analysis
3. Complete s^m symmetrical factorial designs with s=3.
4. Completely confounded s^m symmetrical factorial designs with s=3, m=2.
5. Partially confounded s^m symmetrical factorial designs with s=3, m=2.
6. sk-p fractional factorial designs with s=3, k(≤4).
7. Split-plot designs.

Discipline-Specific Core (DSC) Course- 2b: Stochastic Processes

Course Title & Code	Credits	Credit Distribution of the Course		
		Lecture (45 Hours)	Tutorial (15 Hours)	Practical/ Practice
DSC 2b: Stochastic Processes	4	3	1	0

Course objectives: The main objective of this course is to develop awareness for the use of stochastic models for representing random phenomena evolving in time such as inventory or queueing situations or stock prices behavior

Course Learning Outcomes:

After successful completion of this course, student will be able to:

1. Use notions of long-time behaviour including transience, recurrence, and equilibrium in applied situations such as branching processes and random walk.

2. Construct transition matrices for Markov dependent behaviour and summarize process information
3. Use selected statistical distributions for modeling various phenomena.
4. Understand the principles and objectives of model building based on Markov chains, Poisson processes and Brownian motion.

Unit I: (12 Hours)

Review of Basic Probability Concepts. Introduction to Stochastic Processes. Deterministic and Stochastic Exponential Growth Models. Stationary and Evolutionary Processes. Poisson Processes: Poisson distribution and Poisson Process. Arrival, Interarrival and Conditional Arrival Distributions. Nonhomogeneous Processes. Law of Rare Events and Poisson Process. Poisson Point Process. Distributions associated with Poisson Process. Compound Poisson Processes.

Unit II: (12 Hours)

Markov Chains: Transition Probability Matrices, Chapman- Kolmogorov equations, Some Examples and Classification of States, Regular Chains and Stationary Distributions, Periodicity, Limit theorems. Some Applications. Patterns for recurrent events: One-dimensional, two-dimensional and three-dimensional random walks.

Unit III: (11 Hours)

Brownian Motion: Limit of Random Walk, Its Defining Characteristics and Peculiarities. Its Variations: Standard Brownian Motion, Brownian Bridge, Brownian Motion Reflected at Origin, Geometric Brownian Motion, Brownian Motion with Drift. Reflection Principle. Some Applications.

Unit IV: (10 Hours)

Renewal Processes: Preliminaries, Elementary Renewal Theorem, Delayed Renewal Processes. Limit Theorems. Martingales: Definitions and Some Examples, Stopping Times, Martingale Stopping Theorem, Wald Equation.

Suggested Readings:

1. Bhat, B.R. (2000). Stochastic Models- Analysis and Applications, New Age International Publishers.

2. Feller, William (1968). An Introduction to Probability Theory and its Applications, Vol. 1, 3rd Edn., John Wiley & Sons.
3. Karlin, S. and Taylor, H.M. (1975). A first course in Stochastic Processes, Second ed. Academic Press
4. Medhi, J. (1994). Stochastic Processes, Seconded Wiley Eastern Ltd.
5. Prabhu, N.U. (2007). Stochastic Processes: Basic Theory and its Applications, World Scientific
6. Ross, S. M. (1996). Stochastic Processes, John Wiley and Sons, Inc
7. Taylor, H.M. and Karlin, S. (1998). An Introduction to Stochastic Modelling, 3rd ed., Academic Press.

Teaching Plan:

Week 1:	Review of basic Probability Concepts. Introduction to Stochastic Processes. Deterministic and Stochastic Exponential Growth Models. Stationary and Evolutionary Processes.
Week 2:	Poisson Processes: Poisson Distribution and Poisson Process. Arrival, Interarrival and Conditional Arrival Distributions. Nonhomogeneous Processes.
Week 3:	Poisson Processes: Law of Rare Events and Poisson Process. Poisson Point Process.
Week 4:	Distributions associated with Poisson Process. Compound Poisson Processes.
Week 5:	Markov Chains: Transition Probability Matrices, Chapman- Kolmogorov equations, Some Examples and Classification of States
Week 6:	Markov Chains: Regular Chains and Stationary Distributions, Periodicity
Week 7:	Markov Chains: Limit theorems, Fundamental Matrix, Some Applications
Week 8:	Patterns for recurrent events: One-dimensional, two-dimensional and three-dimensional random walks.
Week 9:	Brownian Motion: Limit of Random Walk, Its Defining Characteristics and Peculiarities.
Week 10:	Brownian Motion: Its Variations: Standard Brownian Motion, Brownian Bridge, Brownian Motion Reflected at Origin, Geometric Brownian Motion, Brownian Motion with Drift.
Week 11:	Brownian Motion: Reflection Principle. Some Applications.
Week 12:	Renewal Processes: Preliminaries, Elementary Renewal Theorem, Delayed Renewal Processes.

Week 13:	Renewal Processes: Limit Theorems. Martingales: Definitions and some examples.
Week 14:	Stopping Times, Martingale Stopping Theorem, Wald Equation.

Discipline-Specific Elective (DSE) Courses

Discipline-Specific Elective (DSE) Courses - 2a: Linear Algebra

Course Title & Code	Credits	Credit Distribution of the Course		
		Lecture (45 Hours)	Tutorial (15 Hours)	Practical/ Practice
DSE 2a: Linear Algebra	**4**	**3**	**1**	**0**

Course Objectives: The main objective of this paper is to allow students to manipulate and understand multidimensional space.

Course Learning Outcomes: After completing this course students will

1. Demonstrate a deep understanding of vector spaces, subspaces, linear independence, basis, and dimension.
2. Analyze and interpret linear transformations, matrix representations, and change of basis, including orthogonality and inner product spaces.
3. Apply orthogonality and the Gram-Schmidt process in practical problems.
4. Compute eigenvalues, eigenvectors, and perform spectral and decomposition.
5. Use generalized inverses and quadratic forms in practical problems.

Unit I: (10 Hours)

Concept of groups and fields with examples, Vector spaces and Subspaces with examples, Direct sum and Algebra of subspaces viz. sum, intersection, union etc, Linear combinations, Spanning sets, Linear spans, Linear dependence and independence in vector spaces, Row and Column space of a matrix, Basis and Dimensions.

Unit II: (11 Hours)

Linear Transformations, Kernel and Image of a linear transformation, Rank and Nullity, Matrix representation of a linear operator, Change of Basis, Similarity, Inner product spaces with examples, Cauchy-Schwarz inequality with applications, Orthogonality, Orthonormal sets and Bases, Gram Schmidt Orthogonalization Process.

Unit III: (12 Hours)

Eigenvalues and eigenvectors, Spectral decomposition of a symmetrical matrix (Full rank and non-full rank cases), Example of spectral decomposition, Spectral decomposition of asymmetric matrix, Cayley Hamilton theorem, Algebraic and geometric multiplicity of characteristic roots, Diagonalization of matrices, Factorization of a matrix, Eigenvalues and eigenvectors for solution of Differential equations.

Unit IV: (12 Hours)

Generalized inverse of a matrix, Different classes of Generic ized inverse, Properties of g-inverse, Reflexive g-inverse, left Weak and right Weak g-inverse, Moore- Penrose (MP) g-inverse and its properties, Real quadratic form, Linear transformation of quadratic forms, Index and signature, Reduction of quadratic form into sum of squares, Gram matrix with example, Jordan canonical form.

Suggested Readings:

1. Biswas, S. (1997). A Text Book of Matrix Algebra, 2nd ed., New Age International Publishers.
2. Golub, G.H. and Van Loan, C.F. (1989). Matrix Computations, 2nd ed., John Hopkins University Press, Baltimore-London.
3. Hadley, G. (2002). Linear Algebra. Narosa Publishing House (Reprint).
4. Rao, C.R. (1973). Linear Statistical Inferences and its Applications, 2nd ed., John Wiley & Sons.
5. Robinson, D.J.S. (1991). A Course in Linear Algebra with Applications, World Scientific, Singapore.
6. Searle, S.R. (1982). Matrix Algebra useful for Statistics, John Wiley & Sons.
7. Strang, G. (1980). Linear Algebra and its Application, 2nd ed., Academic Press, London New York.

Teaching Plan:

Week 1:	Concept of groups and fields with examples, Vector spaces and Subspaces with examples, Direct sum of subspaces.
Week 2:	Algebra of subspaces viz. sum, intersection, union etc, Linear combinations, Spanning sets, Linear spans
Week 3:	Linear dependence and independence in vector spaces, Row and Column space of a matrix, Basis and Dimensions.
Week 4:	Linear Transformations, Kernel and Image of a linear transformation, Rank and Nullity.
Week 5:	Matrix representation of a linear operator, Change of Basis, Similarity.
Week 6:	Inner product spaces with examples, Cauchy-Schwarz inequality with applications.
Week 7:	Orthogonality, Orthonormal sets and Bases, Gram Schmidt Orthogonalization Process.
Week 8:	Eigenvalues and eigenvectors.
Week 9:	Spectral decomposition of a symmetrical matrix (Full rank and non-full rank cases), Example of spectral decomposition, Spectral decomposition of asymmetric matrix, Cayley Hamilton theorem.
Week 10:	Algebraic and geometric multiplicity of characteristic roots, Diagonalization of matrices, Factorization of a matrix, Eigenvalues and eigenvectors for solution of Differential equations
Week 11-12:	Generalized inverse of a matrix, Different classes of Generic ized inverse, Properties of g-inverse, Reflexive g-inverse, left Weak and right Weak g-inverse, Moore - Penrose (MP) g-inverse and its properties.
Week 13-14:	Real quadratic form, Linear transformation of quadratic forms, Index and signature, Reduction of quadratic form into sum of squares, Gram matrix with example, Jordan canonical form.

Discipline-Specific Elective (DSE) Course- 2b: Non-Parametric Inference

Course Title & Code	Credits	Credit Distribution of the Course		
		Lecture (45 Hours)	Tutorial	Practical/ Practice (30 Hours)
DSE 2b: Non-Parametric Inference	**4**	**3**	**0**	**1**

Course Objectives:

This course will provide the ability to learn the fundamentals of the most relevant nonparametric techniques for statistical inference.

Course Learning Outcomes:

After successful completion of this course, student will be able to:

1. Solve hypothesis testing problems where the conditions for the traditional parametric inferential tools to be applied are not fulfilled.
2. Build nonparametric density estimates.

Unit I: (11 Hours)

Review of order statistics, Distribution-free statistics over a class, counting statistics, ranking statistics, Statistics utilizing counting and ranking, Asymptotic distribution of U-statistics, Confidence interval for population quantile and scale parameter, point estimation. Estimators associated with distribution free test statistics, Exact small-sample and asymptotic properties of the Hodges-Lehmann location estimators.

Unit II: (11 Hours)

Nonparametric density estimation, nonparametric regression estimation. Tests based on length of the longest run, runs up and down, Kolmogorov-Smirnov two sample statistic. Rank order statistics: Correlation between ranks and variate values, One sample, paired sample and two sample problems, distribution properties of linear rank statistics.

Unit III: (11 Hours)

Tests for equality of k independent samples: Kruskal-Wallis one way ANOVA test, Measures of Association for bivariate samples: Kendall's Tau coefficient, Spearman's coefficient of Rank correlation, relations between R and T; E (R), τ and ρ.

Unit IV: (12 Hours)

Measures of association in multiple classifications: Friedman's two-way ANOVA by ranks in a k x n table, the Coefficient of Concordance of k sets of rankings of n objects, the Coefficient of Concordance of k sets of incomplete rankings. Concept of power and robustness, elements of bootstrapping.

Suggested Readings:

1. David, H.A. and Nagaraja, H. N. (2003). Order Statistics, Third Edition, John Wiley & Sons.
2. Gibbons, J.D. and Chakraborti, S. (1992). Nonparametric Statistical Inference, 3rd ed., Marcel Dekker.
3. Hettmansperger, T.P. (1984). Statistical inference Based on Ranks, John Wiley & Sons.
4. Randles, R.H. and Wolfe, D.A. (1979). Introduction to the Theory of Nonparametric Statistics, John Wiley & Sons.
5. Rohatgi, V.K. and Saleh, A.K. Md. E. (2005). An Introduction to Probability and Statistics, 2^{nd} ed., John Wiley & Sons.

Teaching Plan:

Week 1:	Review of order statistics, Distribution-free statistics over a class, Counting statistics, ranking statistics, Statistics utilizing counting and ranking.
Week 2:	Asymptotic distribution of U-statistics, Confidence interval for population quantile and scale parameter.
Week 3:	Point estimation. Estimators associated with distribution free test statistics.
Week 4:	Exact small-sample and asymptotic properties of the Hodges-Lehmann location estimators.
Week 5:	Nonparametric density estimation, Nonparametric regression estimation. Tests based on length of the longest run, runs up and down.
Week 6:	Kolmogorov-Smirnov two sample statistic. Rank order statistics: Correlation between ranks and variate values.

Week 7:	One sample, paired sample and two sample problems, distribution properties of linear rank statistics.
Week 8:	Tests for equality of k independent samples: Kruskal-Wallis one way ANOVA test.
Week 9:	Measures of Association for bivariate samples: Kendall's Tau coefficient.
Week 10:	Spearman's coefficient of Rank correlation, relations between R and T; E (R), τ and ρ.
Week 11:	Measures of association in multiple classifications: Friedman's two-way ANOVA by ranks in a k x n table.
Week 12-13:	Coefficient of Concordance of k sets of rankings of n objects, Coefficient of Concordance of k sets of incomplete rankings.
Week 14:	Concept of power and robustness, elements of bootstrapping.

List of Practicals:

1. Point estimation. Estimators associated with distribution free test statistics.
2. Kolmogorov-Smirnov two sample statistic.
3. Correlation between ranks and variate values
4. One sample, paired sample and two sample problems.
5. Kruskal-Wallis one way ANOVA test.
6. Kendall's Tau coefficient.
7. Spearman's coefficient of Rank correlation, relations between R and T; E (R), τ and τ.
8. Friedman's two-way ANOVA by ranks in a k x n table.
9. Coefficient of Concordance of k sets of rankings of n objects.
10. Coefficient of Concordance of k sets of incomplete rankings.

Discipline-Specific Elective (DSE) Course- 2c: Statistical Quality Control

Course Title & Code	Credits	Credit Distribution of the Course		
		Lecture (45 Hours)	Tutorial	Practical/ Practice (30 Hours)
DSE 2c: Statistical Quality Control	4	3	0	1

Course Objectives: The main purpose of this paper is to introduce the most important field of applied statistics that contributes to quality control in almost all industries.

Course Learning Outcomes:

After successful completion of this course, student will be able to:

1. Describe the DMAIC process (define, measure, analyze, improve, and control).
2. Demonstrate to use the methods of statistical process control and to determine when an out-of-control situation has occurred.
3. Design and use Cumulative sum chart, tabular Cumulative sum chart and V-mask schemes for detecting small shifts of the mean from goal conditions.
4. Choose an appropriate sampling inspection technique.
5. Gain the ability to understand the concept of errors in making inference
6. Understand the concept of OC and ARL of control chart.
7. Understand the concept of Dodge's continuous sampling inspection plans.

Unit I: (11 Hours)

Basic concepts of process monitoring and process control, Generic theory and review of attributes and variable control charts, errors in making inferences from control charts, OC and ARL of control charts.

Unit II: (11 Hours)

Moving average and exponentially weighted moving average control chart (EWMA), cumulative sum control chart (CUSUM) using V-mask and decision intervals, economic design of $\overline{X}$ Chart.

Unit III: (12 Hours)

Methods and philosophy of statistical process control, process and measurement system capability analysis: process capability ratios, process capability analysis using a control chart, gauge and measurement system capability studies.

Unit IV: (11 Hours)

Review of sampling inspection techniques, single sampling plans, double sampling plans, multiple sampling plans, sequential sampling plans and their properties, Dodge's continuous sampling inspection plans for inspection by variables for one-sided and two-sided specifications.

Suggested Readings:

1. Biswas, S. (1996). Statistics of Quality Control, Sampling Inspection and Reliability, New Age International Publishers.
2. Duncan A.J. (1974). Quality Control and Industrial Statistics, IV Edision, Taraporewala and Sons.
3. Ott, E. R. (1977). Process Quality Control (McGraw Hill) Montgomery, D. C. (2005).
4. Introduction to Statistical Quality Control, 5^{th} ed., John Wiley & Sons.
5. Wetherill, G. B. (1977). Sampling Inspection and Quality Control, Halsted Press.
6. Wetherill, G.B. Brown, D.W. (1991). Statistical Process Control Theory and Practice, Chapman & Hall.

Teaching Plan:

Week 1:	Basic concepts of process monitoring and process control, Generic theory and review of attributes
Week 2:	Review of variable control charts, errors in making inferences from control charts, OC and ARL of control charts.
Week 3:	Moving average and exponentially weighted moving average control chart (EWMA)
Week 4:	cumulative sum control chart (CUSUM) using V-mask and decision intervals
Week 5:	Economic design of $\overline{X}$ Chart.
Week 6:	Methods and philosophy of statistical process control
Week 7:	Process and measurement system capability analysis: process capability ratios
Week 8:	process capability analysis using a control chart
Week 9:	Gauge and measurement system capability studies

Week 10:	Review of sampling inspection techniques, single sampling plans
Week 11:	Double sampling plans, multiple sampling plans
Week 12:	Sequential sampling plans and their properties
Week 13:	Dodge's continuous sampling inspection plans for inspection by variables for one-sided specifications
Week 14:	Dodge's continuous sampling inspection plans for inspection by variables for two-sided specifications.

List of Practicals: Students will be required to do practical based on topics listed below

1. Control charts for mean and range
2. Control charts for mean and standard deviation
3. Control charts for individual units
4. Lot-by-lot attribute sampling plans
5. Cumulative sum control chart
6. Moving average control chart
7. Exponentially weighted moving average control chart
8. Process capability analysis procedure.

Discipline-Specific Elective (DSE) Course – 2d: Reliability Theory

Course Title & Code	Credits	Credit Distribution of the Course		
		Lecture (45 Hours)	Tutorial (15 Hours)	Practical/ Practice
DSE 2d: Reliability Theory	4	3	1	0

Course Objectives:

1. Introduces fundamental reliability concepts, including reliability, availability, and maintainability, and explores their interrelationships.
2. Define reliability and explain its significance in engineering and systems.
3. Learn key reliability measures such as failure rate, mean time to failure (MTTF), and mean time between failures (MTBF).

4. Understand redundancy techniques and their applications in improving system reliability.
5. Understand new better than used (NBU), decreasing mean residual life (DMRL), and new better than used in expectation (NBUE) properties.

Course Learning Outcomes:

1. To analyze the system reliability, including coherent systems and their reliability estimation.
2. Investigate reliability in systems with imperfect switches and priority redundant systems.
3. Explain the loss of memory property of the exponential distribution and its significance in failure modeling.
4. Apply Markov models to reliability function analysis and use regenerative point techniques to analyze system reliability.

Unit I: (12 Hours)

Reliability concepts & measures, components and systems, coherent system, reliability of the coherent system. Availability, types of availability- point wise, interval, asymptotic. Failure rate and mean time to failure and their inter-relationships. Statistical failure models: exponential. gamma, weibull, pareto, normal, lognormal and related distributions.

Unit II: (10 Hours)

System components and configurations: series-parallel, parallel-series, and K-out of –N-system. Series- strength reliability and its estimation. Reliability Bounds- classical and Bayesian approach.

Unit III: (11 Hours)

Maintenance Policies, System with imperfect switch. Concept of redundancy, types of redundancy, priority redundant, repairable system, comparison of component, unit and standby redundancies.

Unit IV: **(12 Hours)**

Model Plotting techniques: Reliability function with Markov model. Two unit cold standby & parallel unit system with constant failure rate, Arbitrary Repair rates & Regenerative point Techniques. Stress strength reliability and its estimation. IFR, IFRA, NBU, DMRL and NBUE and their duals, loss of memory property of the exponential distribution.

Suggested Readings:

1. Bain, L.J. and Engelhardt, M. (1991). Statistical Analysis of Reliability and Life- Testing Models, Marcel Dekker Inc., U.S.A.
2. Martz, H.F. and Wailer, R.A. (1982). Bayesian Reliability Analysis, John Wiley and Sons, Inc., New York.
3. Sinha, S.K. (1986). Reliability and Life-Testing, Wiley Eastern Ltd., New Delhi.

Teaching Plan:

Week 1-2:	Reliability concepts & measures, components and systems, coherent system, reliability of the coherent system.
Week 3:	Availability, Types of Availability- point wise, interval, asymptotic. Failure rate and mean time to failure and their inter-relationships. Statistical failure models: exponential. Gamma, Weibull, Pareto, normal, lognormal and related distributions.
Week 4:	System components and configurations: Series, Parallel, Series-Parallel, Parallel-Series, and K-out of –N- system.
Week 5:	Series- strength reliability and its estimation. Redundancy: Types of Redundancies, Repairable system
Week 6:	Maintenance Policies, System with imperfect switch, Priority Redundant system.
Week 7:	Model Plotting techniques: Reliability function with Markov model
Week 8:	Two unit cold standby & parallel unit system with constant failure rate, Arbitrary Repair rates.
Week 9:	Regenerative point Techniques. Stress strength reliability and its estimation. IFR, IFRA, NBU, DMRL and NBUE and their duals
Week 10:	Loss of memory property of the exponential distribution.
Week 11:	Practical on Measuring failure rates of electrical/hypothetical components using historical data.
Week 12:	Practical on Estimating Mean Time to Failure (MTTF) and Mean Time between Failures (MTBF).

Week 13:	Practical on Evaluating series, parallel, and mixed configurations using reliability block diagrams.
Week 14:	Practical on Evaluating cost-benefit analysis of preventive vs corrective maintenance strategies.

Discipline-Specific Elective (DSE) Course - 2e: Computational Techniques

Course Title and Code	Credits	Credit Distribution of Course		
		Lecture (30 Hours)	Tutorial	Practical (60 Hours)
DSE 2e: Computational Techniques	**4**	**2**	**0**	**2**

Course Objectives: The main objective of this course is to allow the students to learn the advanced techniques of modelling real data from diverse discipline.

Course Learning Outcomes:

After successful completion of this course, the students will be able to:

1. Simulate statistical models.
2. Understand linear models and distinguish between fixed, random and mixed effects models.
3. Learn and apply regression technique in their area of study.
4. Understand and apply time series models.

Unit I: (7 Hours)

Probability Distributions: Bernoulli, Binomial, Poisson, Multinomial, Uniform, Exponential, Gamma, Beta, Normal, Chi Square, t and F distribution. Simulation: Random number generation, simulating statistical models, Monte Carlo Methods.

Unit II: (8 Hours)

Linear Models: Fixed, random and mixed effects models, ANOVA: one way and two way, ANOCOVA. Regression Models: Simple and Multiple Linear Regression, Forward, Backward and stepwise regression, Residual analysis. Diagnostics and tests for violations of model assumptions: Multicollinearity, Autocorrelation and Homoscedasticity.

Unit III: (7 Hours)

Generalized Linear Model: Exponential family of distributions, Link function, Canonical link Function, deviance, Logit and Probit models, Logistic and Poisson regression. Lack of fit tests.

Unit IV: (8 Hours)

Time Series: Stationary and Nonstationary time series, Autocorrelation and Auto-covariance functions and their properties, Tests for trend and seasonality. Stationary processes: Moving average (MA) process, Auto-regressive (AR) process, ARMA, ARIMA and SARIMA models. Estimation of mean, auto-covariance and auto-correlation function under large sample theory, forecasting.

Note: Data analysis and applications of the methods are to be carried out using a statistical package like Excel/R/SPSS/MINITAB/MATLAB or any other.

Suggested Readings:

1. Agresti, A. (2015). Foundations of Linear and Generic ized Linear Models, John Wiley, New Jersey.
2. Chatterjee, S. and Hadi, A.S. (2012). Regression Analysis by Example, 5th ed., John Wiley, New Jersey.
3. Cryer, J.D. and Chan, K. (2008). Time Series Analysis: With Applications in R, Springer, New York.
4. Fox, J. and Weisberg, S. (2011). An R Companion to Applied Regression, 2nd ed., Sage.
5. Kroese, D.P. and Chan, J.C.C. (2014). Statistical Modeling and Computation, Springer.
6. Montgomery, D.C. (2001). Designs and Analysis of Experiments, John Wiley & Sons, New York.
7. Montgomery, D.C., Jennings, C.L. and Kulahci, M. (2008). Introduction to Time Series Analysis and Forecasting, John Wiley, New Jersey.
8. Ross, S.M. (2006). Simulation, 4th ed., American Press, USA.
9. Voss, J. (2014). An Introduction to Statistical Computing, John Wiley.
10. Weisberg, S. (2014). Applied Linear Regression, 4th ed., John Wiley,

Teaching Plan:

Week 1:	Probability Distributions: Bernoulli, Binomial, Poisson, Multinomial, Uniform, Exponential, Gamma, Beta, Normal, Chi Square, t and F distribution.
Week 2-3:	Simulation: Random number generation, simulating statistical models, Monte Carlo Methods.
Week 4-5:	Linear Models: Fixed, random and mixed effects models, ANOVA: one was and two way, ANOCOVA.
Week 6-7:	Regression Models: Simple and Multiple Linear Regression, Forward, Backward and stepwise regression, Residual analysis. Diagnostics and tests for violations of model assumptions: Multi-co-linearity, Autocorrelation and Homoscedasticity.
Week 8-10:	Generalized Linear Model: Exponential Family of distributions, Link function, Canonical link Function, deviance, Logit and Probit models, Logistic and Poisson regression, Lack of fit tests.
Week 11:	Time Series: Stationary and nonstationary time series, Autocorrelation and Auto-covariance functions and their properties, Tests for trend and seasonality.
Week 12-14:	Stationary processes: Moving average (MA) process, Auto-regressive (AR) process, ARMA, ARIMA and SARIMA models. Box-Jenkins model, Estimation of mean, auto-covariance and auto-correlation function under large sample theory, forecasting.

List of Practicals:

1. Fitting of Probability distributions.
2. Random number generation
3. Problem based on one way ANOVA and Two-way ANOVA
4. Problem based on ANOCOVA
5. Fitting of Linear Regression, Forward, Backward and stepwise regression.
6. Logistic and Poisson regression. Lack of fit tests.
7. Time Series: Tests for trend and seasonality.
8. Fitting and forecasting of various time series processes: MA, AR, ARMA, ARIMA, SARIMA.

Generic Elective (GE) COURSES

Generic Elective (GE) Course- 2a: Statistics for Research and Management Studies

Course Title & Code	Credits	Credit Distribution of the Course		
		Lecture (30 Hours)	Tutorial	Practical/ Practice (60 Hours)
GE 2b: Statistics for Research and Management Studies	4	2	0	2

Course Objectives: The main objectives of this course are:

1. To learn statistical techniques useful for research work.
2. To understand the quantitative methods used in business and management studies.

Course Learning Outcomes: After completing this course, the students will be able to:

1. Know different types of data produced in their area of study.
2. Create, manage, visualize, and summarize datasets.
3. Use and understand the inferential procedures.
4. Apply suitable sampling design.
5. Understand and apply basic designs.
6. Apply regression techniques.
7. Apply suitable statistical techniques to analyze the data and interpret the results.

Unit I: (7 Hours)

Data types, scale of measurement, creating and managing datasets, importing and exporting data, data cleaning. Summarizing data: Frequency and probability distributions, measures of central tendency, measures of dispersion, skewness and kurtosis. Correlation and regression, Measures of association, Cross tabulation. Visualizing data: Histogram, bar chart, pie chart, stem and leaf display, scatter plot, box and whisker plot.

Unit II: (8 Hours)
Inference: Population and sample, parameter and statistic, estimates and estimators, estimation of parameters, testing of hypothesis, type I and type II errors, p-value, inferences based on sample. Tests based on sampling distributions: Z, t, χ^2 and F.

Unit III: (8 Hours)
Concept of population and sample, complete enumeration versus sampling, sampling and non-sampling errors. Types of sampling: non-probability and probability sampling, basic principles of sample survey, Sampling Techniques: Simple random sampling, stratified random sampling, Cluster Sampling, Systematic sampling.

Unit IV: (7 Hours)
ANOVA for one way and two-way classification. ANOCOVA, analysis of basic designs, analysis of 2 level factorial experiments. Simple and multiple regression, logistic regression.

Note: Data analysis and applications of the methods are to be carried out using calculator or by using a statistical package like Excel/R.

Suggested Readings:

1. Agresti, A. (2015). Foundations of Linear and Generic ized Linear Models, John Wiley, New Jersey.
2. Cochran, W.G. (2011). Sampling Techniques, 3rd ed., Wiley Eastern John Wiley & Sons.
3. Cochran, W.G. (2011): Sampling Techniques (3rd Ed.), Wiley Eastern John Wiley and Sons.
4. DeGroot, M.H. and Schervish, M.J. (2012). Probability and Statistics, 4th ed., Pearson Education.
5. Field, A., Miles, J. and Field, Z. (2012). Discovering Statistics using R, Sage, Los Angels.
6. Judd, C. M., McClelland, G. H. and Ryan, C.S. (2009). Data Analysis: A Model Comparison Approach, 2nd ed., Routledge, New York.
7. Montgomery, D.C. (2001). Design and Analysis of Experiments, 5th ed., John Wiley, New
8. Raghava Rao, D. (1988). Exploring Statistics, Markel Dekker, New York.
9. Rice, J.A. (1995). Mathematical Statistics and Data Analysis, 2nd ed., Duxbury Press.

10. Sukhatme, P. V., Sukhatme, B. V., Sukhatme, S., Asok, C. (1984). Sampling Theories of Survey with Application, IOWA State University Press and Indian Society of Agricultural Statistics.
11. Taylor, J.K. and Cihon, C. (2004). Statistical Techniques for Data Analysis, 2nd ed., Chapman & Hall.

Teaching Plan:

Week 1-3:	Data types, scale of measurement, creating and managing dataset, importing and exporting data, data cleaning. Summarizing data: Frequency and probability distributions, measures of central tendency, measures of dispersion, skewness and kurtosis. Correlation and regression, Measures of association, Cross tabulation. Visualizing data: Histogram, bar chart, pie chart, stem and leaf display, scatter plot, box and whisker plot.
Week 4-6:	Inference: Population and sample, parameter and statistic, estimates and estimators, estimation of parameters, type I and type II errors, p-value, statistical hypotheses, testing of hypothesis, inferences based on sample. Tests based on sampling distributions: Z, t, χ^2 and F.
Week 7-10:	Sampling Techniques: Simple random sampling, stratified random sampling, Cluster Sampling, Systematic sampling.
Week 11-14:	ANOVA for one way and two-way classification. ANOCOVA, analysis of basic designs, analysis of 2 level factorial experiments. Simple and multiple regression, logistic regression.

List of Practicals:

Students will be required to do practical based on topics listed below:

1. Measures of central tendency, measures of dispersion, skewness and kurtosis.
2. Correlation and regression, measures of association, cross tabulation.
3. One-way ANOVA and two-way ANOVA and ANOCOVA.
4. Histogram, bar chart, pie chart, stem and leaf display, scatter plot, box and whisker plot
5. Testing of hypothesis.
6. Different sampling techniques as per syllabus.
7. Tests based on sampling distributions: Z, t, χ^2 and F.

Skill Based/Specialized Laboratory Courses

Skill Based/Specialized Laboratory Courses- 2a: Data Analysis Using Python

Course Title & Code	Credits	Credit Distribution of the Course		
		Lecture	Tutorial	Practical/ Practice (60 Hours)
SB 2a: Data Analysis Using Python	2	0	0	2

Course Objective:

This course introduces Python as a tool for statistical analysis, covering data visualization, probability, statistical inference, regression, sampling, and matrix algebra. Students will learn to fit probability distributions, conduct hypothesis testing, and apply statistical models to real-world data using key Python libraries

Course Learning Outcomes:

1. Understand Python programming for statistical analysis
2. Develop computational skills for probability, inference, regression, and matrix algebra
3. Perform hypothesis testing, distribution fitting and ANOVA
4. Apply regression techniques and model selection for statistical analysis

Software & Libraries Required:

1. Python (Jupyter Notebook / Google Colab)
2. Libraries: NumPy, Pandas, SciPy, Statsmodels, Seaborn, Matplotlib, Scikit-learn

Unit 1: (15 Hours)

Python basics: Variables, loops, functions, list comprehensions, NumPy: One-dimensional & two-dimensional arrays, operations, Pandas: Data Frames, filtering, grouping, handling missing data, Data visualization: Histograms, boxplots, scatterplots, KDE plots, bar charts, pie charts, Probability distributions & simulations: Binomial, Poisson, Normal, Exponential, Monte Carlo methods, Law of Large Numbers & Central Limit Theorem.

Unit 2: (15 Hours)

Estimation: MLE, confidence intervals, bootstrapping, Hypothesis testing: Z-test, t-test, F-test, Wilcoxon signed-rank test, Kruskal-Wallis test, Analysis of variance (ANOVA): One-way, two-way, post-hoc tests (Tukey's HSD), Interpretation & visualization of hypothesis tests and ANOVA results.

Unit 3: (15 Hours)

Simple & multiple linear regression, assumptions, residual analysis and diagnostic plots, Model selection: Adjusted R^2, AIC, BIC, stepwise selection. Survey sampling: Simple random sampling, stratified sampling, estimation of population parameters. Handling missing data & graphical representation of survey results. Markov Chains: transition probability matrix, steady-state probabilities. Simulation of Markov Chains in Python.

Unit 4: (15 Hours)

Matrix operations: Addition, multiplication, inversion, eigenvalues, eigenvectors Spectral decomposition & solving linear systems. Fitting probability distributions using MLE & SciPy.stats Goodness-of-Fit Tests: Kolmogorov-Smirnov, Anderson-Darling, Chi-Square Visualization of fitted distributions using histograms & QQ-plots.

Suggested Readings:

1. Balagurusamy, E. (2016). Introduction to computing and problem solving using python. McGraw Hill Education, First Edition.
2. Halswanter, T. (2016). An Introduction to Statistics with Python: With Applications in the Life Sciences. Springer.
3. McKinney, W. (2012). Python for data analysis: Data wrangling with Pandas, NumPy, and IPython. " O'Reilly Media, Inc.".
4. Perkovic, L. (2015). Introduction to computing using python: An application development focus. John Wiley & Sons.
5. Vander Plas, J. (2016). Python data science handbook: Essential tools for working with data. " O'Reilly Media, Inc.

Teaching Plans:

Week 1:	Introduction to Python, Jupyter/Colab, NumPy basics.
Week 2:	Pandas for data handling, filtering, grouping, and EDA.
Week 3:	Data visualization with Matplotlib & Seaborn, customizing plots.
Week 4:	Probability distributions, Monte Carlo simulations, LLN & CLT.

Week 5:	Estimation (MLE, confidence intervals), hypothesis testing (Z-test, t-test).
Week 6:	Advanced tests (F-test, ANOVA, Wilcoxon, Kruskal-Wallis), post-hoc analysis.
Week 7-8:	Regression analysis, model assumptions, selection (AIC, BIC, Adjusted R^2), diagnostic plots.
Week 9:	Sampling techniques, population estimation, handling missing data.
Week 10:	Markov Chains, transition probabilities, steady-state, simulation.
Week 11-12:	Matrix algebra: operations, inverse and g-inverses, eigenvalues, solving linear systems.
Week 13–14:	Probability distribution fitting, Goodness-of-Fit tests, visualization.

DEPARTMENT OF STATISTICS
FACULTY OF MATHEMATICAL SCIENCES
UNIVERSITY OF DELHI, DELHI - 110007

M.A./M.Sc. (Statistics)

First Year of Two Year PG Programme - Level 6

Semester	Discipline-Specific Core (DSC) Courses	Discipline-Specific Elective (DSE) Courses	Genric Elective	Skill Based/ Specialized Laboratory
I	1a Probability Theory 1b Statistical Methodology 1c Survey Sampling	1a Analysis 1b Time Series Analysis 1c Biostatistics 1d Official and National Development Statistics	1a Official and National Development Statistics 1b Statistical Computing Using R	1a Data Analysis Using Excel
II	2a Statistical Inference 2b Design of Experiments 2c Stochastic Processes	2a Linear Algebra 2b Non-Parametric Inference 2c Statistical Quality Control 2d Reliability Theory 2e Computational Techniques	2a Computational Techniques 2b Statistics for Research and Management Studies	2a Data Analysis Using Python

Second Year of Two Year PG Programme - Level 6.5

Semester	Discipline-Specific Core (DSC) Courses	Discipline-Specific Elective (DSE) Courses	Genric Elective	Skill Based/ Specialized Laboratory
III	3a Advanced Statistical Inference 3b Multivariate Analysis	3a Statistics in Finance 3b Statistical Quality Management 3c Advanced Survey Sampling 3d Advanced Theory of Experimental Designs 3e Operational Research 3f Acturial Statistics 3g Stochastic Models	3a Actuarial Statistics 3b Essentials of Survey Sampling and Experimental Design* 3c Applied Multivariate Statistics	3a Data Analysis Using R
IV	4a Generalized Linear Models 4b Econometrics	4a Bayesian Inference 4b Order Statistics 4c Applied Stochastic Processes 4d Advanced Statistical Computing and Data Mining 4e Statistical Decision Theory 4f Forestry and Environmental Statistics	4a Forestry and Environmental Statistics 4b Inferential Techniques	4a Data Analysis Using SPSS

Annexure-5

Two-year M.Sc. Mathematics

(based on Post Graduate Curriculum Framework (PGCF) -2024)

Effective from Academic Session 2025-26

Department of Mathematics

Faculty of Mathematical Sciences
University of Delhi
Delhi - 110007

CONTENTS

Programme Objectives and Outcomes

Programme Objectives

The M.Sc. Mathematics programme's main objectives are to:

- inculcate and develop mathematical aptitude and the ability to think abstractly in the students.
- develop computational abilities and programming skills of the students.
- develop the ability to read, follow and appreciate mathematical text among the students.
- train students to communicate mathematical ideas in a lucid and effective manner.
- train students to apply their theoretical knowledge to solve real-life problems.
- prepare the students for higher education and research in mathematics.

Programme Outcomes

On successful completion of the M.Sc. Mathematics programme, a student will:

- have a strong foundation in core areas of Mathematics, both pure and applied.
- be able to apply mathematical skills and logical reasoning for problem solving.
- communicate mathematical ideas effectively, in writing as well as orally.
- have sound knowledge of mathematical modelling, programming and computational techniques as required for employment in industry.

Curricular Structure for First Year

Semester	DSC	DSE	2 Credit Course	Total Credits
Semester-I	**DSC-1** **DSC-2** **DSC-3** (12 Credits)	**DSE-1** **DSE-2** OR **DSE-1 & GE-1** (8 Credits)	**Skill-based course/ Workshop/ Specialized laboratory/ Hands on learning** (2 Credits)	**22**
Semester-II	**DSC-4** **DSC-5** **DSC-6** (12 Credits)	**DSE-3** **DSE-4** OR **DSE-2 & GE-2** (8 Credits)	**Skill-based course/ Workshop/ Specialized laboratory/ Hands on learning** (2 Credits)	**22**

Details of Courses in 1st Year of Two-year M.Sc. Mathematics

Semester	DSC	DSE	2 Credit Course
Semester-I	**DSC-1:** Field Theory **DSC-2**: Introduction to Topology **DSC-3**: Ordinary Differential Equations	**DSE-1:** (i) Matrix Analysis (ii) Numerical Analysis **DSE-2*:** (i) Advanced Group Theory (ii) Nonlinear Optimization *Student will opt for DSE-2 or GE-1	Communicating Mathematics
Semester-II	**DSC-4**: Module Theory **DSC-5**: Functional Analysis **DSC-6**: Complex Analysis	**DSE-3:** (i) Algebraic Number Theory (ii) General Topology **DSE-4#:** (i) Fourier Analysis (ii) Integral Equations #Student will opt for DSE-4 or GE-2	Appreciating Mathematics via Workshops and Seminars

GE Courses (offered by the Department of Mathematics)

Semester-I	**GE-1:** (i) Matrix Analysis (ii) Nonlinear Optimization
Semester-II	**GE-2**: (i) Fourier Analysis (ii) Integral Equations

Semester-I

Discipline Specific Core (DSC) Courses

DISCIPLINE SPECIFIC CORE – 1: FIELD THEORY

CREDIT DISTRIBUTION OF THE COURSE

Course Title & Code	Credits	Credit Distribution of the Course			Eligibility Criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical		
DSC-1: Field Theory	4	3	1	0	Same as for entry to M.Sc. Mathematics	Basics of Groups and Rings

Learning Objectives

The primary objective of this course is to:

- study the ancient problem of solvability of polynomials over the rational field.
- understand a necessary and sufficient condition for the solvability of polynomials in terms of radical expression.
- apply Galois theory to classical problems, such as the insolvability of the general quintic.

Learning Outcomes

This course will enable the students to:

- identify and construct examples of fields, distinguish between algebraic and transcendental extensions, characterize normal extensions in terms of splitting fields and prove the existence of algebraic closure of a field.
- characterize perfect fields using separable extensions, construct examples of automorphism group of a field and Galois extensions as well as prove Artin's theorem and the fundamental theorem of Galois theory.
- classify finite fields using roots of unity and Galois theory and prove that every finite separable extension is simple.
- use Galois theory of equations to prove that a polynomial equation over a field of characteristic is solvable by radicals if and only if its group (Galois) is a solvable group and hence deduce that a general quintic equation is not solvable by radicals.

SYLLABUS OF DSC-1

Unit – 1 **(12 hours)**

Fields and their extensions, Splitting fields, Normal extensions, Algebraic closure of a field.

Unit – 2 **(12 hours)**

Separability, Perfect fields, Automorphisms of field extensions, Artin's theorem, Galois extensions, Fundamental theorem of Galois theory.

Unit – 3 **(10 hours)**

Roots of unity, Cyclotomic polynomials and extensions, Finite fields, Theorem of primitive element and Steinitz's theorem.

Unit – 4 **(11 hours)**

Galois theory of equations, Theorem on natural irrationalities, Radical extension and solvability by radicals.

Tutorial

Problem-solving sessions based on material covered in the lectures will be taken up in the tutorials along with scholastic work related to conceptual understanding of the subject.

Essential Readings

[1] P. M. Cohn, *Basic Algebra*, Springer International Edition, 2003.

Suggested Readings

(i) D. S. Dummit & R. S. Foote, *Abstract Algebra*, Wiley Student Edition, 2011.
(ii) T. W. Hungerford, *Algebra*, Springer-Verlag, 1981.
(iii) N. Jacobson, *Basic Algebra*, Volume I, Dover Publications Inc., 2009.
(iv) I. Stwart, *Galois Theory*, CRC Press, 2015.

DISCIPLINE SPECIFIC CORE – 2: INTRODUCTION TO TOPOLOGY

CREDIT DISTRIBUTION OF THE COURSE

Course Title & Code	Credits	Credit Distribution of the Course			Eligibility Criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical		
DSC-2: Introduction to Topology	4	3	1	0	Same as for entry to M.Sc. Mathematics	Knowledge of Metric Spaces

Learning Objectives

The primary objective of this course is to introduce:

- basic principles of point-set topology, including bases and subbases for a topology.
- continuity, homeomorphisms, and different types of topologies, such as product and box topologies.
- key notions of connectedness and local connectedness.
- compactness and its significance in topological spaces.

Learning Outcomes

This course will enable the students to:

- analyze subsets of topological spaces by determining their interior, closure, boundary, and limit points, as well as identifying bases and subbases.
- identify continuous functions between topological spaces, analyze mappings into product spaces, and compare topological properties of different spaces.
- evaluate the connectedness and path connectedness of the product of an arbitrary family of spaces.
- understand key classifications of topological spaces, including Hausdorff spaces, first and second countable spaces, and separable spaces.
- explore advanced concepts such as limit point compactness and Tychonoff's theorem.

SYLLABUS OF DSC-2

Unit – 1 **(10 hours)**
Topological spaces, Basis, Order topology, Subspace topology, Metric topology, Closed set and limit points, Hausdorff spaces.

Unit – 2 **(12 hours)**
Continuous functions, Homeomorphism, The box and product topologies, Metrizability of products of metric spaces, Connected and path connected spaces.

Unit – 3 **(12 hours)**
Locally connected and locally path connected spaces, Connectedness of product of spaces, First and second countable spaces, Separable spaces.

Unit – 4 **(11 hours)**
Compact spaces, The Tychonoff theorem, Limit point compactness, Sequential compactness.

Tutorial

Problem-solving sessions based on material covered in the lectures will be taken up in the tutorials along with scholastic work related to conceptual understanding of the subject.

Essential Readings

[1] J. R. Munkres, *Topology*, Updated Second Edition, Pearson, 2021.
[2] T. B. Singh, *Introduction to Topology*, Springer Nature, 2019.

Suggested Readings

(i) G. E. Bredon, *Topology and Geometry*, Springer, 2014.
(ii) J. Dugundji, *Topology*, Allyn and Bacon Inc., Boston, 1978.
(iii) J. L. Kelley, *General Topology*, Dover Publications, 2017.
(iv) G. F. Simmons, *Introduction to Topology and Modern Analysis*, McGraw Hill Education, 2017.
(v) L. A. Steen & J. A. Seebach, *Counterexamples in Topology*, Dover Publications, 2013.
(vi) S. Willard, *General Topology*, Dover Publications, 2004.

DISCIPLINE SPECIFIC CORE – 3: ORDINARY DIFFERENTIAL EQUATIONS

CREDIT DISTRIBUTION OF THE COURSE

Course Title & Code	Credits	Credit Distribution of the Course			Eligibility Criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical		
DSC-3: Ordinary Differential Equations	**4**	**3**	**1**	**0**	**Same as for entry to M.Sc. Mathematics**	**Knowledge of Differential Equations and Calculus**

Learning Objectives

The objective of this course is to study:

- existence, uniqueness, and continuity of solutions of initial value problems (IVPs)
- homogeneous and non-homogeneous linear systems
- stability of solutions for systems of ordinary differential equations.
- eigenvalues and eigenfunctions of Sturm-Liouville systems and Green's functions
- applications of theory of ordinary differential equations in real world problems.

Learning Outcomes

After studying this course, the student will be able to:

- know about the existence, uniqueness, and continuity of solutions of IVPs.
- apply the matrix method of solution for linear systems of differential equations.
- analyze the stability of solutions for systems of ordinary differential equations.
- understand Green's functions and their applications in the solution of boundary value problems (BVPs).
- comprehend the properties of eigenvalues and eigenfunctions of Sturm-Liouville systems.

SYLLABUS OF DSC-3

Unit – 1 **(12 hours)**

Well-posed problems, Existence, uniqueness, and continuity theorems for the solution of IVPs of the first order, Picard's method, Existence and uniqueness of solution for systems and higher order IVPs , Global existence theorem.

Unit – 2 **(9 hours)**

Homogeneous and non-homogeneous linear systems, Linear systems with constant coefficients and their solution by matrix method, Linear equations with periodic coefficients.

Unit – 3 **(12 hours)**

Stability of autonomous system of differential equations, Critical points of an autonomous system and their classification. Stability of linear systems with constant coefficients, Linear plane autonomous system and phase portrait analysis, Perturbed systems, Method of Lyapunov for nonlinear systems, Limit cycles, Poincare-Bendixson's theorem and its applications.

Unit – 4 **(12 hours)**

Sturm separation and comparison theorems, Adjoint forms and Lagrange's identity, Two-point boundary value problems, Green's functions, Construction of Green's functions, Sturm-Liouville systems, eigenvalues and eigenfunctions.

Tutorial

Problem-solving sessions based on material covered in the lectures will be taken up in the tutorials along with scholastic work related to conceptual understanding of the subject.

Essential Readings

[1] E. A. Coddington, *An Introduction to Ordinary Differential Equations*, Dover Publications, 2012.
[2] T. Myint-U, *Ordinary Differential Equations,* Elsevier, North-Holland, 1978.
[3] S. L. Ross, *Differential Equation*s, Second Edition, John Wiley & Sons, India, 2007.

Suggested Readings

(i) L. Perko, *Differential Equations and Dynamical Systems,* Springer, 2001.
(ii) G. F. Simmons, *Differential Equations with Applications and Historical Notes*, Third Edition, CRC Press, 2017.

Discipline Specific Elective (DSE) Courses

DISCIPLINE SPECIFIC ELECTIVE – 1 (i): MATRIX ANALYSIS

CREDIT DISTRIBUTION OF THE COURSE

Course Title & Code	Credits	Credit Distribution of the Course			Eligibility Criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical		
DSE-1 (i): Matrix Analysis	4	3	1	0	Same as for entry to M.Sc. Mathematics	Knowledge of Linear Algebra

Learning Objectives

The primary objective of this course is to:

- create a bridge between undergraduate-level matrix theory and matrix theory used in applications and research.
- give exposure to some topics of linear algebra that find application in analysis.

Learning Outcomes

This course will enable the students to learn:

- various useful matrix decompositions.
- variety of matrix norms and useful facts about the spectral radius.
- positive definite matrices and matrix ordering.
- some useful products of matrices.
- the powerful tool of non-negative matrices and the celebrated Perron's theorem which arises in many applications.

SYLLABUS OF DSE-1 (i)

Unit – 1 **(10 hours)**

Schur decomposition, Spectral decomposition, Jordan decomposition, Singular value decomposition, QR factorization, LU factorization.

Unit – 2 **(12 hours)**

Matrix norms, Gelfand formula for spectral radius, Matrix functions including matrix exponential, Gershgorin regions, Condition number of matrix, Eigenvalue perturbation theorem, Bauer-Fike's theorem.

Unit – 3 **(12 hours)**

Positive definite and semidefinite matrices and their properties, A pair of positive semidefinite matrices (matrix-ordering), Polar value decomposition, Fischer's inequality, Hadamard's inequality, Minkowski's inequality.

Unit – 4 **(11 hours)**

Schur and Kroneckar products and their properties, Schur and Kroneckar products of positive definite matrices, Permutation and Doubly Stochastic matrices, Birkhoff's theorem, Non-negative matrices and their properties, Perron's theorem.

Tutorial

Problem-solving sessions based on material covered in the lectures will be taken up in the tutorials along with scholastic work related to conceptual understanding of the subject.

Essential Readings

[1] R. A. Horn & C. R. Johnson, *Matrix Analysis*, Second Edition, Cambridge University Press, 2013.
[2] F. Zhang, *Matrix Theory: Basic Results and Techniques*, Second Edition, Springer, 2011.

Suggested Readings

(i) R. Bhatia, *Matrix Analysis*, Springer, 1997.
(ii) A. J. Laub, *Matrix Analysis for Scientists and Engineers*, SIAM, 2005.
(iii) C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*, SIAM, 2000.

DISCIPLINE SPECIFIC ELECTIVE – 1 (ii): NUMERICAL ANALYSIS

CREDIT DISTRIBUTION OF THE COURSE

Course Title & Code	Credits	Credit Distribution of the Course			Eligibility Criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical		
DSE-1 (ii): Numerical Analysis	4	3	1	0	Same as for entry to M.Sc. Mathematics	Knowledge of Calculus and Real Analysis

Learning Objectives

The main objective of this course is to:

- get robust understanding of the basic theory behind numerical methods.
- analyze the accuracy, efficiency, and stability of numerical algorithms.
- develop an ability to evaluate the suitability of various numerical techniques based on problem context and computational constraints.

Learning Outcomes

This course will enable the students to:

- apply algorithms for solving linear and nonlinear equations, analyze the convergence properties and potential pitfalls of these methods.
- develop a comprehensive understanding of the core concepts and principles behind both direct and indirect solvers for system of linear equations.
- use various interpolation techniques and understand their error estimates.
- implement various techniques to approximate derivatives and integrals and analyze them.
- construct and analyze numerical methods for solving initial value problems, covering key aspects like convergence, stability, and error propagation.

SYLLABUS OF DSE-1 (ii)

Unit – 1 **(11 hours)**

Floating-point approximation of a number, Source and propagation of error, Stability in numerical computation, Polynomial deflation, Laguerre's method, Horner's method, Muller's method for real and complex roots, Their order of convergence and convergence analysis.

Unit – 2 **(12 hours)**

Existence and uniqueness of interpolating polynomial, Interpolation error, Neville's method of interpolation, Cubic spline interpolation, Trigonometric interpolation, Data fitting.

Unit – 3 **(12 hours)**

Least square approximation and Chebyshev approximation, Special Matrices: Diagonal dominant matrices, Positive definite matrices, Band matrices and Tridiagonal matrices and their important properties, Eigen value problem by using Power method, Adaptive quadrature, Gaussian integration methods.

Unit – 4 **(10 hours)**

Initial value problems, Difference equations, Local and global truncation error, Convergence and stability, Single step and multi-step methods, First and fourth-order Taylor series method, Extrapolation method, Adams-Moulton and Adams-Bashforth methods, Nystrom method.

Tutorial

Problem-solving sessions based on material covered in the lectures will be taken up in the tutorials along with scholastic work related to conceptual understanding of the subject.

Essential Readings

[1] K. E. Atkinson, *An Introduction to Numerical Analysis*, Second Edition, Wiley-India, 1989.
[2] R. L. Burden & J. D. Faires, *Numerical Analysis*, Ninth Edition, Cengage Learning India Private Limited, 2012.
[3] S. D. Conte & C. de Boor, *Elementary Numerical Analysis - An Algorithmic Approach*, Third Edition, McGraw-Hill, 1981.

Suggested Readings

(i) B. Bradie, *A Friendly Introduction to Numerical Analysis*, Pearson Education, India, 2006.
(ii) C. F. Gerald & P. O. Wheatley, *Applied Numerical Analysis*, Seventh Edition, Pearson Education, 2004.
(iii) F. B. Hildebrand, *Introduction to Numerical Analysis*, Second Edition, Dover Publications, 2008.

DISCIPLINE SPECIFIC ELECTIVE – 2 (i): ADVANCED GROUP THEORY

CREDIT DISTRIBUTION OF THE COURSE

Course Title & Code	Credits	Credit Distribution of the Course			Eligibility Criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical		
DSE-2 (i): Advanced Group Theory	4	3	1	0	Same as for entry to M.Sc. Mathematics	Basics of Group Theory

Learning Objectives
The objective of this course is to introduce:
- the concepts of normal series, composition series and Zessenhaus lemma.
- solvable groups, nilpotent group and fitting and Frattini subgroup.
- free group, presentation of a group and properties of a free group.

Learning Outcomes
This course will enable the students to:
- understand Schreier's refinement theorem, Jordan-Hölder theorem, and fundamental theorem of arithmetic using Jordan-Hölder theorem.
- learn the significance and proof of Hall's theorem, Schur's theorem and Burnside basis theorem.
- identify indecomposable spaces and to prove the Krull-Schmidt theorem.
- determine distinct presentations of a group.

SYLLABUS OF DSE-2 (i)

Unit – 1 **(12 hours)**
Normal series, Composition series, Zessenhaus lemma, Schreier's refinement theorem, Jordan-Hölder theorem.

Unit – 2 **(11 hours)**
Solvable groups, Derived series, Supersolvable groups, Minimal normal subgroup, Hall's theorem, Hall subgroup, p-complements, Central series, Schur's theorem.

Unit – 3 **(11 hours)**
Nilpotent groups, Fitting subgroup, Jacobi identity, Three subgroup lemma, Frattini subgroup, Burnside basis theorem.

Unit – 4 **(11 hours)**
Indecomposable groups, Fitting's lemma, Krull-Schmidt theorem, Semidirect product, Free group, Generators and relations of a group.

Tutorial
Problem-solving sessions based on material covered in the lectures will be taken up in the tutorials along with scholastic work related to conceptual understanding of the subject.

Essential Readings

[1] J. J. Rotman, *An Introduction to the Theory of Groups*, Springer-Verlag, New York, 1995.

Suggested Readings

(i) T. W. Hungerford, *Algebra*, Springer-Verlag, New York, 1981.
(ii) D. J. S. Robinson, *A Course in the Theory of Groups*, Springer-Verlag, New York, 1996.
(iii) J. S. Rose, *A Course on Group Theory*, Dover Publication, New York, 1994.
(iv) M. Suzuki, *Group Theory-I*, Springer-Verlag, Berlin, 1982.

DISCIPLINE SPECIFIC ELECTIVE – 2 (ii): NONLINEAR OPTIMIZATION

CREDIT DISTRIBUTION OF THE COURSE

Course Title & Code	Credits	Credit Distribution of the Course			Eligibility Criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical		
DSE-2 (ii): Nonlinear Optimization	**4**	**3**	**1**	**0**	**Same as for entry to M.Sc. Mathematics**	**Knowledge of Calculus, Real Analysis, Two and Three Dimensional Geometry**

Learning Objectives

The primary objective of this course is to introduce:

- convex functions and separation theorems.
- optimality conditions for unconstrained and constrained nonlinear optimization problems.
- Lagrangian duals and study duality results.
- Wolfe's method for quadratic programming problems.

Learning Outcomes

This course will enable the students to:

- derive first and second order optimality conditions for unconstrained optimization problems.
- know the importance of Karush–Kuhn–Tucker necessary optimality conditions in constrained optimization.
- understand duality theory and saddle point theory in terms of Lagrangian function.
- investigate saddle point theory.

SYLLABUS OF DSE-2 (ii)

Unit – 1 **(10 hours)**

Existence theorems, First order optimality conditions for unconstrained optimization problems, Second order optimality conditions for unconstrained optimization problems.

Unit – 2 **(10 hours)**

Convex functions, Differentiable convex functions, Optimization on convex sets, Optimality conditions, Separation theorems.

Unit – 3 **(13 hours)**

Fritz John optimality conditions for constrained optimization problems, Constraint qualifications, Karush−Kuhn−Tucker conditions for constrained optimization problems, Second order necessary and sufficient optimality conditions.

Unit – 4 **(12 hours)**

Langragian function, Lagrangian duality in nonlinear optimization, Strong duality in convex programming, Saddle points, Saddle point optimality, Duality for linear and quadratic problems, Wolfe's method for solving quadratic programming problems.

Tutorial

Problem-solving sessions based on material covered in the lectures will be taken up in the tutorials along with scholastic work related to conceptual understanding of the subject.

Essential Readings

[1] H. A. Eiselt & Carl-Louis Sandblom, *Nonlinear Optimization: Methods and Applications*, Springer, 2019.
[2] O. Güler, *Foundations of Optimization*, Springer, 2010.

Suggested Readings

(i) M. S. Bazaraa, H. D. Sherali & C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*, Third Edition, John Wiley & Sons, 2013.
(ii) D. G. Luenberger & Y. Ye, *Linear and Nonlinear Programming*, Springer, 2008.
(iii) A. P. Ruszczyński, *Nonlinear Optimization*, Princeton University Press, 2006.

Two-Credit Course

COMMUNICATING MATHEMATICS

Course Title	Credits	Credit Distribution of the Course			Eligibility Criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical		
Communicating Mathematics	**2**	**2**	**0**	**0**	**Same as for entry to M.Sc. Mathematics**	**NIL**

Learning Objectives

This course will train the students to read mathematics independently and to present it adeptly via posters as well as oral presentations.

Learning Outcomes

The course will enable the students to effectively communicate mathematical ideas, both verbally and in writing. It will also hone their analytical skills and their ability to think critically while encouraging them to work collaboratively.

Methodology

A group of 4-5 students will be assigned a mathematical article/ paper published in reputed journals/ periodicals, like *The American Mathematical Monthly* (Mathematical Association of America, Taylor and Francis, Link: https://www.tandfonline.com/journals/uamm20), *Mathematics Magazine* (Mathematical Association of America, Taylor and Francis, Link: https://www.tandfonline.com/journals/umma20), *The Mathematics Student* (Indian Mathematical Society, Link: https://www.indianmathsoc.org/MS.html) and *The Mathematical Intelligencer* (Springer Nature, Link: https://link.springer.com/journal/283). The designated groups will be required to comprehend these articles under the supervision of a faculty member. The poster and oral presentations will be conducted for these groups as a part of their assessment process.

Generic Elective (GE) Courses

GENERIC ELECTIVE – 1 (i): MATRIX ANALYSIS

CREDIT DISTRIBUTION OF THE COURSE

Course Title & Code	Credits	Credit Distribution of the Course			Eligibility Criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical		
GE-1 (i): Matrix Analysis	4	3	1	0	Class XII pass with Mathematics	Knowledge of Basic Analysis and Linear Algebra

Learning Objectives

The primary objective of this course is to:

- create a bridge between undergraduate-level matrix theory and matrix theory used in applications and research.
- give exposure to some topics of linear algebra that find application in analysis.

Learning Outcomes

This course will enable the students to learn:

- various useful matrix decompositions.
- variety of matrix norms and useful facts about the spectral radius.
- positive definite matrices and matrix ordering.
- some useful products of matrices.
- powerful tool of non-negative matrices and the celebrated Perron's theorem which arises in many applications.

SYLLABUS OF GE-1 (i)

Unit – 1 **(10 hours)**

Schur decomposition, Spectral decomposition, Jordan decomposition, Singular value decomposition, QR factorization, LU factorization.

Unit – 2 **(12 hours)**

Matrix norms, Gelfand formula for spectral radius, Matrix functions including matrix exponential, Gershgorin regions, Condition number of matrix, Eigenvalue perturbation theorem, Bauer-Fike's theorem.

Unit – 3 **(12 hours)**

Positive definite and semidefinite matrices and their properties, A pair of positive semidefinite matrices (matrix-ordering), Polar value decomposition, Fischer's inequality, Hadamard's inequality, Minkowski's inequality.

Unit – 4 **(11 hours)**

Schur and Kroneckar products and their properties, Schur and Kroneckar products of positive definite matrices, Permutation and Doubly Stochastic matrices, Birkhoff's theorem, Non-negative matrices and their properties, Perron's theorem.

Tutorial

Problem-solving sessions based on material covered in the lectures will be taken up in the tutorials along with scholastic work related to conceptual understanding of the subject.

Essential Readings

[1] R. A. Horn & C. R. Johnson, *Matrix Analysis*, Second Edition, Cambridge University Press, 2013.
[2] F. Zhang, *Matrix Theory: Basic Results and Techniques*, Second Edition, Springer, 2011.

Suggested Readings

(i) R. Bhatia, *Matrix Analysis*, Springer, 1997.
(ii) A. J. Laub, *Matrix Analysis for Scientists and Engineers*, SIAM, 2005.
(iii) C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*, SIAM, 2000.

GENERIC ELECTIVE – 1 (ii): NONLINEAR OPTIMIZATION

CREDIT DISTRIBUTION OF THE COURSE

Course Title & Code	Credits	Credit Distribution of the Course			Eligibility Criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical		
GE-1 (ii): Nonlinear Optimization	4	3	1	0	Class XII pass with Mathematics	Knowledge of Calculus, Real Analysis, Two and Three Dimensional Geometry

Learning Objectives

The primary objective of this course is to introduce:
- convex functions and separation theorems.
- optimality conditions for unconstrained and constrained nonlinear optimization problems.
- Lagrangian duals and study duality results.
- Wolfe's method for quadratic programming problems.

Learning Outcomes

This course will enable the students to:
- derive first and second order optimality conditions for unconstrained optimization problems.
- know the importance of Karush–Kuhn–Tucker necessary optimality conditions in constrained optimization.
- understand duality theory and saddle point theory in terms of Lagrangian function.
- investigate saddle point theory.

SYLLABUS OF GE-1 (ii)

Unit – 1 **(10 hours)**
Existence theorems, First order optimality conditions for unconstrained optimization problems, Second order optimality conditions for unconstrained optimization problems.

Unit – 2 **(10 hours)**
Convex functions, Differentiable convex functions, Optimization on convex sets, Optimality conditions, Separation theorems.

Unit – 3 **(13 hours)**
Fritz John optimality conditions for constrained optimization problems, Constraint qualifications, Karush−Kuhn−Tucker conditions for constrained optimization problems, Second order necessary and sufficient optimality conditions.

Unit – 4 **(12 hours)**
Langragian function, Lagrangian duality in nonlinear optimization, Strong duality in convex programming, Saddle points, Saddle point optimality, Duality for linear and quadratic problems, Wolfe's method for solving quadratic programming problems.

Tutorial

Problem-solving sessions based on material covered in the lectures will be taken up in the tutorials along with scholastic work related to conceptual understanding of the subject.

Essential Readings

[1] H. A. Eiselt & Carl-Louis Sandblom, *Nonlinear Optimization: Methods and Applications*, Springer, 2019.
[2] O. Güler, *Foundations of Optimization*, Springer, 2010.

Suggested Readings

(i) M. S. Bazaraa, H. D. Sherali & C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*, Third Edition, John Wiley & Sons, 2013.
(ii) D. G. Luenberger & Y. Ye, *Linear and Nonlinear Programming*. Springer, 2008.
(iii) A. P. Ruszczyński, *Nonlinear Optimization*, Princeton University Press, 2006.

Semester-II

Discipline Specific Core (DSC) Courses

DISCIPLINE SPECIFIC CORE – 4: MODULE THEORY

CREDIT DISTRIBUTION OF THE COURSE

Course Title & Code	Credits	Credit Distribution of the Course			Eligibility Criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical		
DSC-4: Module Theory	4	3	1	0	Same as for entry to M.Sc. Mathematics	Basics of Groups and Rings

Learning Objectives
The primary objective of this course is to:
- introduce a new algebraic structure, namely, module which is a generalization of a vector space when the underlying field is replaced by an arbitrary ring. The study of modules over a ring also provides an insight into the structure of the ring.
- study free modules, finitely generated modules, projective and injective modules.
- classify the finitely generated modules over a principal ideal domain (PID).

Learning Outcomes
This course will enable the students to:
- identify and construct examples of modules, and apply homomorphism theorems on the same.
- define and characterize Noetherian, Artinian module, and apply the structure theorem of finitely generated modules over PID.
- distinguish between projective, injective, free, and semi simple modules.
- prove universal property of tensor product of modules, and Hilbert basis theorem.

SYLLABUS OF DSC-4

Unit – 1 **(13 hours)**
Basic concepts of module theory, Fundamental theorems of homomorphism, Direct product and direct sum of modules, Exact sequences, Split exact sequences.

Unit – 2 **(10 hours)**
Free modules, Projective and injective modules, Dual basis lemma, Baer's criterion, Divisible modules.

Unit – 3 **(12 hours)**
Tensor product of modules, Chain conditions, Hilbert basis theorem.

Unit – 4 **(10 hours)**
Modules over PID's, Semi simple modules.

Tutorial

Problem-solving sessions based on material covered in the lectures will be taken up in the tutorials along with scholastic work related to conceptual understanding of the subject.

Essential Readings

[1] M. F. Athiyah & I. G. Macdonald, *Introduction to Commutative Algebra*, Addison Wesley, 1969.
[2] P. M. Cohn, *Basic Algebra*, Springer International Edition, 2003.
[3] P. M. Cohn, *Classic Algebra*, John Wiley & Sons Ltd., 2000.

Suggested Readings

(i) D. S. Dummit & R. M. Foote, *Abstract Algebra*, Wiley India Pvt. Ltd., 2011.
(ii) N. Jacobson, *Basic Algebra*, Volume II, Dover Publications Inc., 2009.
(iii) T. W. Hungerford, *Algebra*, Springer-Verlag, 1981.

DISCIPLINE SPECIFIC CORE – 5: FUNCTIONAL ANALYSIS

CREDIT DISTRIBUTION OF THE COURSE

Course Title & Code	Credits	Credit Distribution of the Course			Eligibility Criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical		
DSC-5: Functional Analysis	4	3	1	0	Same as for entry to M.Sc. Mathematics	Knowledge of Metric Spaces and Basic Linear Algebra and Analysis

Learning Objectives

The primary objective of this course is to:

- familiarize students with the basic tools of Functional Analysis involving normed spaces, Banach spaces and Hilbert spaces.
- investigate various properties of continuous and linear transformations.
- introduce different types of bounded linear operators.

Learning Outcomes

This course will enable the students to:

- understand completeness with respect to a norm and the interplay between continuity and boundedness of linear operators.
- comprehend and apply Hahn-Banach theorem, Open mapping theorem, Closed graph theorem and the Uniform boundedness theorem.
- understand the decomposition of a Hilbert space in terms of orthogonal complements and representation of a bounded linear functional in terms of inner product.
- check the convergence of operators and functionals, and weak and strong convergence of sequences.

SYLLABUS OF DSC-5

Unit – 1 **(12 hours)**

Banach spaces, Linear and continuous operators, Normed spaces of operators, Dual spaces.

Unit – 2 **(13 hours)**

Hahn-Banach theorem, Consequences of the Hahn-Banach theorem, Natural imbedding map, Open mapping theorem, Closed graph theorem, Uniform boundedness theorem.

Unit – 3 **(12 hours)**

Hilbert spaces, Orthogonal complements, Orthonormal sets, Reflexivity of Hilbert spaces, Hilbert-adjoint of an operator.

Unit – 4 **(8 hours)**

Conjugate of an operator, Self-adjoint operators, Normal and unitary operators, Projections, Eigen values, spectrum and resolvent of an operator.

Tutorial

Problem-solving sessions based on material covered in the lectures will be taken up in the tutorials along with scholastic work related to conceptual understanding of the subject.

Essential Readings

[1] G. F. Simmons, *Introduction to Topology and Modern Analysis*, McGraw Hill Education, 2017.

Suggested Readings

(i) G. Bachman & L. Narici, *Functional Analysis*, Dover Publications, 2000.
(ii) R. Bhatia, *Notes on Functional Analysis,* Hindustan Book Agency, India, 2009.
(iii) E. Kreyszig, *Introductory Functional Analysis with Applications*, John Wiley & Sons, India, 2006.
(iv) M. Schechter, *Principles of Functional Analysis,* Second Edition, American Mathematical Society, 2001.

DISCIPLINE SPECIFIC CORE – 6: COMPLEX ANALYSIS

CREDIT DISTRIBUTION OF THE COURSE

Course Title & Code	Credits	Credit Distribution of the Course			Eligibility Criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical		
DSC-6: Complex Analysis	4	3	1	0	Same as for entry to M.Sc. Mathematics	Knowledge of Real Analysis and Metric Spaces

Learning Objectives

The primary objective of this course is to:

- gain insights of well-known classical results in the field of complex analysis.
- investigate various properties of analytic functions, conformal mappings and Möbius transformations.
- derive various forms of Cauchy's theorem, integral formulas and maximum principles.
- represent complex-valued functions as Taylor and Laurent series.

Learning Outcomes

This course will enable the students to:

- construct Möbius transformations using Symmetry and Orientation principles.
- foresee the usage of simply connected regions in the complex plane for the existence of primitives and branch of logarithm.
- understand the behavior of zeros of analytic functions and meromorphic functions through Argument principle and Rouche's theorem.
- evaluate the real integrals involving rational and trigonometric functions by contour integration using Residue theorem.
- apply Schwarz's lemma to characterize the conformal maps of the open unit disk onto itself.

SYLLABUS OF DSC-6

Unit – 1 **(10 hours)**

Extended plane and its spherical representation, Analytic functions, Branch of logarithm, Conformal mappings, Möbius transformations.

Unit – 2 **(10 hours)**

Line integrals, Fundamental theorem of Calculus for line integrals, Power series representation of analytic functions, Zeros of analytic functions, Liouville's theorem.

Unit – 3 **(12 hours)**

Index of a closed curve, Cauchy's theorem and integral formula, Morera's Theorem, Homotopic version of Cauchy's theorem and simple connectivity, Counting Zeros, Open mapping theorem, Goursat's theorem.

Unit – 4 **(13 hours)**

Classification of singularities, Laurent series, Casorati-Weierstrass theorem, Residue theorem with applications, Argument principle, Rouche's theorem, Maximum principles, Schwarz lemma.

Tutorial
Problem-solving sessions based on material covered in the lectures will be taken up in the tutorials along with scholastic work related to conceptual understanding of the subject.

Essential Readings

[1] J. B. Conway, *Functions of One Complex Variable*, Second Edition, Narosa Publishing House, New Delhi, 2002.

Suggested Readings

(i) L. V. Ahlfors, *Complex Analysis*, Mc Graw Hill Co., Indian Edition, 2017.
(ii) T. W. Gamelin, *Complex Analysis*, Springer New York, NY, 2001.
(iii) L. Hahn & B. Epstein, *Classical Complex Analysis*, Jones and Bartlett, 1996.
(iv) E. M. Stein & R. Shakarchi, *Complex Analysis*, Princeton University Press, 2003.

Discipline Specific Elective (DSE) Courses

DISCIPLINE SPECIFIC ELECTIVE – 3 (i): ALGEBRAIC NUMBER THEORY

CREDIT DISTRIBUTION OF THE COURSE

Course Title & Code	Credits	Credit Distribution of the Course			Eligibility Criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical		
DSE-3 (i): Algebraic Number Theory	**4**	**3**	**1**	**0**	**Same as for entry to M.Sc. Mathematics**	**Knowledge of Groups, Rings and Fields**

Learning Objectives

The primary objective of this course is to introduce:

- algebraic number fields and their ring of integers.
- factorization into irreducibles in the ring of integers.
- factorization of ideals of the ring of integers.
- lattices, geometric representation of algebraic numbers, class-group and class-number.

Learning Outcomes

This course will enable the students to:

- classify algebraic number fields, define algebraic integers, ring of integers and integral bases, and calculate norms and traces. It would be possible to determine the integral bases and ring of integers of quadratic and p-th cyclotomic fields.
- construct examples of non-unique factorization domains and apply the unique factorization of certain ring of integers of number fields to solve some Diophantine equations.
- prove uniqueness of factorization of ideal of ring of integers of a number field in terms of prime ideals. It also leads to deduction of Two-Squares and Four-Squares theorem using Minkowski's theorem on convex sets.
- visualize ideals of the ring of integers as lattices and develop tools to prove the finiteness of class-group.

SYLLABUS OF DSE – 3 (i)

Unit – 1 **(11 hours)**

Algebraic numbers, Conjugates and discriminants, Algebraic integers, Integral bases, Norms and traces, Rings of algebraic integers.

Unit – 2 **(12 hours)**

Quadratic and cyclotomic fields, Trivial factorization, Factorization into irreducibles, Examples of non-unique factorization into irreducibles, Prime factorization, Euclidean domains, Euclidean quadratic fields, Consequence of unique factorization the Ramanujan-Nagell theorem.

Unit – 3 **(12 hours)**

Prime factorization of ideals, Norm of an ideal, Lattices, Quotient torus, Minkowski's theorem, Two-Squares theorem, Four-Squares theorem.

Unit – 4 **(10 hours)**

The space L^{st}, Class-group and class-number, Finiteness of the class-group, Factorization of a rational prime, Minkowski's constants, Some class-number calculations.

Tutorial

Problem-solving sessions based on material covered in the lectures will be taken up in the tutorials along with scholastic work related to conceptual understanding of the subject.

Essential Readings

[1] I. Stewart & D. Tall, *Algebraic Number Theory and Fermat's Last Theorem*, Fourth Edition, CRC Press, Boca Raton, FL, 2016.

Suggested Readings

(i) Ş. Alaca & K. S. Williams, *Introductory Algebraic Number Theory*, Cambridge University Press, Cambridge, 2003.
(ii) K. Ireland & M. Rosen, *A Classical Introduction to Modern Number Theory*, Second Edition, GTM 84, Springer-Verlag, New York, 1990.
(iii) S. Lang, *Algebraic Number Theory*, Second Edition, GTM 110, Springer-Verlag, New York, 1994.
(iv) D. A. Marcus, *Number Fields*, Second Edition, Universitext, Springer, 2018.

DISCIPLINE SPECIFIC ELECTIVE – 3 (ii): GENERAL TOPOLOGY

CREDIT DISTRIBUTION OF THE COURSE

Course Title & Code	Credits	Credit Distribution of the Course			Eligibility Criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical		
DSE-3 (ii): General Topology	4	3	1	0	Same as for entry to M.Sc. Mathematics	DSC-2: Introduction to Topology

Learning Objectives
The primary objective of this course is to introduce:
- quotient spaces, local compactness, and one-point compactification.
- separation axioms, the Urysohn lemma, and the Tietze extension theorem.
- paracompactness, metrization theorems, and the partition of unity.

Additionally, the course aims to equip students with essential tools and foundational knowledge for conducting advanced research in topology and related fields.

Learning Outcomes
This course will enable the students to:
- explore notable examples of quotient spaces, such as cones and suspensions.
- determine the one-point compactification of spaces like the real line and the n-sphere.
- understand key results on complete regularity and the Stone-Čech compactification.
- study fundamental theorems, including Urysohn's lemma and Tietze extension theorem.
- learn about important metrization theorems, such as the Urysohn metrization theorem.
- gain insights into characterizations of paracompactness in regular spaces and the role of partition of unity.

SYLLABUS OF DSE – 3 (ii)

Unit – 1 **(10 hours)**
Quotient spaces, Identification maps, Local compactness, One-point compactification.

Unit – 2 **(12 hours)**
Proper Maps and regularity, Complete regularity, Stone-Čech compactification.

Unit – 3 **(11 hours)**
Normality, Urysohn's lemma, Tietze extension theorem, Urysohn metrization theorem.

Unit – 4 **(12 hours)**
Paracompactness, Characterizations of paracompactness in regular spaces, Partition of unity.

Tutorial
Problem-solving sessions based on material covered in the lectures will be taken up in the tutorials along with scholastic work related to conceptual understanding of the subject.

Essential Readings
[1] J. R. Munkres, *Topology*, Updated Second Edition, Pearson, 2021.

[2] T. B. Singh, *Introduction to Topology*, Springer Nature, 2019.

Suggested Readings

(i) G. E. Bredon, *Topology and Geometry*, Springer, 2014.
(ii) J. Dugundji, *Topology*, Allyn and Bacon Inc., Boston, 1978.
(iii) J. L. Kelley, *General Topology*, Dover Publications, 2017.
(iv) G. F. Simmons, *Introduction to Topology and Modern Analysis*, McGraw Hill Education, 2017.
(v) L. A. Steen & J. A. Seebach, *Counterexamples in Topology*, Dover Publications, 2013.
(vi) S. Willard, *General Topology*, Dover Publications, 2004.

DISCIPLINE SPECIFIC ELECTIVE – 4 (i): FOURIER ANALYSIS

CREDIT DISTRIBUTION OF THE COURSE

Course Title & Code	Credits	Credit Distribution of the Course			Eligibility Criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical		
DSE-4 (i): Fourier Analysis	4	3	1	0	Same as for entry to M.Sc. Mathematics	Knowledge of Real Analysis including Riemann Integration

Learning Objectives
The primary objective of this course is to introduce:
- basic tools related to Fourier series and Fourier multipliers.
- time-localized and frequency-localized signals.
- finite Fourier transforms and see their applications.
- time-frequency localized bases and filter banks.
- some types of summability kernels.

Learning Outcomes
This course will enable the students to:
- derive Fourier inversion formula for functions in finite dimensional spaces.
- calculate the finite Fourier transform including Parseval's identity.
- comprehend the translation invariance of operations including the convolution product.
- realize the role of Fourier multipliers in signal analysis.
- explore time and frequency-localized signals.
- analyze discrete signals in terms of time-frequency localized bases.
- understand summability kernels and Fourier coefficients.

SYLLABUS OF DSE-4 (i)

Unit – 1 **(13 hours)**
Properties of the finite Fourier transform: The Fourier inversion formula, Parseval's identity, Computation of the finite Fourier transform, Finite Fourier transform and translation-invariant linear operator, Circulant matrices.

Unit – 2 **(12 hours)**
Basic properties of the convolution operator, Translation invariance of the convolution product, Fourier multipliers, Relation between the convolution operator and Fourier multipliers.

Unit – 3 **(10 hours)**
Time-frequency analysis: Time and frequency-localized signals, time-localized bases, frequency-localized bases, involution, properties of involution, computation of discrete signals.

Unit – 4 **(10 hours)**
Cesàro summation of series, Riemann Lebesgue lemma, Fourier series, Dirichlet's and Fejér's kernels, Uniqueness theorem, Fourier coefficients of derivatives, Pointwise convergence.

Tutorial
Problem-solving sessions based on material covered in the lectures will be taken up in the tutorials

along with scholastic work related to conceptual understanding of the subject.

Essential Readings

[1] A. Vretblad, *Fourier Analysis and its Application*, Springer-Verlag, New York, 2003.
[2] M. W. Wong, *Discrete Fourier Analysis*, Birkhäuser, 2011.

Suggested Readings

(i) S. A. Broughton & K. Bryan, *Discrete Fourier Analysis and Wavelets*, Second Edition, John Wiley & Sons, Inc., 2018.
(ii) V. Serov, *Fourier Series, Fourier Transform and their Applications to Mathematical Physics*, Springer, 2017.

DISCIPLINE SPECIFIC ELECTIVE – 4 (ii): INTEGRAL EQUATIONS

CREDIT DISTRIBUTION OF THE COURSE

Course Title & Code	Credits	Credit Distribution of the Course			Eligibility Criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical		
DSE-4 (ii): Integral Equations	4	3	1	0	Same as for entry to M.Sc. Mathematics	Knowledge of Calculus and Differential Equations

Learning Objectives

The main objective of this course is to introduce the learner to:

- the concepts of integral and integro-differential equations.
- methods for solving Volterra and Fredholm integral equations.
- study of non-linear and singular integral equations.
- solutions of integro-differential equations and system of integral equations.

Learning Outcomes

This course will enable the students to:

- compute solutions to Volterra integral equations by different methods.
- solve the system of integral equations and integro-differential equations.
- determine the solutions of Fredholm integral equations and derivation of the Hilbert-Schmidt theorem.
- solve non-linear and singular integral equations.
- relate the integral equations with Green's function.

SYLLABUS OF DSE-4 (ii)

Unit – 1 **(12 hours)**

Types of integral equations, Introduction and relation with linear differential equation, Volterra integral equations and its solutions, Method of resolvent kernels, Method of successive approximations, Decomposition method, Convolution type of equation, Method of Laplace transform.

Unit – 2 **(8 hours)**

System of Volterra integral equations, Solutions of Integro-differential equation, Abel's integral equation and its generalizations, Non-linear integral equations.

Unit – 3 **(12 hours)**

Fredholm integral equations and its solutions, Method of resolvent kernels, Method of successive approximations, Integral equations with degenerate kernels, Solutions of Fredholm integral equations using characteristic numbers and eigenfunctions with their properties.

Unit – 4 **(13 hours)**

Hilbert-Schmidt theorem, Non-homogeneous Fredholm integral equation with symmetric kernel, Fredholm alternatives, Applications of Green's function for solution of the boundary value problems, Singular integral equations, Applications of integral equations: Volterra's population growth model.

Tutorial

Problem-solving sessions based on material covered in the lectures will be taken up in the tutorials along with scholastic work related to conceptual understanding of the subject.

Essential Readings

[1] W. Hackbusch, *Integral Equations: Theory and Numerical Treatment*, Birkhäuser, 1995.
[2] M. L. Krasnov, A. I. Kiselev & G. I. Makarenko, *Problems and Exercises in Integral Equations*, Mir Publication Moscow, 1971.
[3] A. C. Pipkin, *A Course on Integral Equations*, Springer, 1991.
[4] A. M. Wazwaz, *Linear and Non-linear Integral Equations*, Springer 2011.

Suggested Readings

(i) S. G. Georgiev, *Integral Equations on Time Scales*, Atlantis Press, 2016.
(ii) F. B. Hildebrand, *Methods of Applied Mathematics*, Dover Publications, 1992.
(iii) J. D. Logan, *Applied Mathematics*, Fourth Edition, John Wiley & Sons, 2013.

Two-Credit Course

APPRECIATING MATHEMATICS VIA WORKSHOPS AND SEMINARS

Course Title	Credits	Credit Distribution of the Course			Eligibility Criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical		
Appreciating Mathematics via Workshops and Seminars	**2**	**2**	**0**	**0**	**Same as for entry to M.Sc. Mathematics**	**NIL**

Learning Objectives

In this course, students will be trained to grasp mathematical ideas and techniques disseminated in workshops and seminars.

Learning Outcomes

This course will enable the students to quickly distill the central ideas in any mathematical discourse, particularly outside the classroom setting, and write a short summary encompassing these ideas.

Methodology

The students will attend a requisite number of workshops and seminars organized by the Department through the semester. They will be expected to prepare and submit summaries of a few of these (as mandated by the Department) for assessment.

Generic Elective (GE) Courses

GENERIC ELECTIVE – 2 (i): FOURIER ANALYSIS

CREDIT DISTRIBUTION OF THE COURSE

Course Title & Code	Credits	Credit Distribution of the Course			Eligibility Criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical		
GE-2 (i): Fourier Analysis	4	3	1	0	Class XII pass with Mathematics	Knowledge of Real Analysis including Riemann Integration

Learning Objectives

The primary objective of this course is to introduce:

- basic tools related to Fourier series and Fourier multipliers.
- time-localized and frequency-localized signals.
- finite Fourier transforms and see their applications.
- time-frequency localized bases and filter banks.
- some types of summability kernels.

Learning Outcomes

This course will enable the students to:

- derive Fourier inversion formula for functions in finite dimensional spaces.
- calculate the finite Fourier transform including Parseval's identity.
- comprehend the translation invariance of operations including the convolution product.
- realize the role of Fourier multipliers in signal analysis.
- explore time and frequency-localized signals.
- analyze discrete signals in terms of time-frequency localized bases.
- understand summability kernels and Fourier coefficients.

SYLLABUS OF GE-2 (i)

Unit – 1 **(13 hours)**

Properties of the finite Fourier transform: The Fourier inversion formula, Parseval's identity, Computation of the finite Fourier transform, Finite Fourier transform and translation-invariant linear operator, Circulant matrices.

Unit – 2 **(12 hours)**

Basic properties of the convolution operator, Translation invariance of the convolution product, Fourier multipliers, Relation between the convolution operator and Fourier multipliers.

Unit – 3 **(10 hours)**

Time-frequency analysis: Time and frequency-localized signals, time-localized bases, frequency-localized bases, involution, properties of involution, computation of discrete signals.

Unit – 4 **(10 hours)**
Cesàro summation of series, Riemann Lebesgue lemma, Fourier series, Dirichlet's and Fejér's kernels, Uniqueness theorem, Fourier coefficients of derivatives, Pointwise convergence.

Tutorial
Problem-solving sessions based on material covered in the lectures will be taken up in the tutorials along with scholastic work related to conceptual understanding of the subject.

Essential Readings
[1] A. Vretblad, *Fourier Analysis and Its Application*, Springer-Verlag, New York, 2003.
[2] M. W. Wong, *Discrete Fourier Analysis*, Birkhäuser, 2011.

Suggested Readings
(i) S. A. Broughton & K. Bryan, *Discrete Fourier Analysis and Wavelets,* Second Edition, John Wiley & Sons, Inc., 2018.
(ii) V. Serov, *Fourier Series, Fourier Transform and their Applications to Mathematical Physics,* Springer, 2017.

GENERIC ELECTIVE – 2 (ii): INTEGRAL EQUATIONS

CREDIT DISTRIBUTION OF THE COURSE

Course Title & Code	Credits	Credit Distribution of the Course			Eligibility Criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical		
GE-2 (ii): Integral Equations	4	3	1	0	Class XII pass with Mathematics	Knowledge of Calculus and Differential Equations

Learning Objectives

The main objective of this course is to introduce the learner to:

- the concepts of integral and integro-differential equations.
- methods for solving Volterra and Fredholm integral equations.
- study of non-linear and singular integral equations.
- solutions of integro-differential equations and system of integral equations.

Learning Outcomes

This course will enable the students to:

- compute solutions to Volterra integral equations by different methods.
- solve the system of integral equations and integro-differential equations.
- determine the solutions of Fredholm integral equations and derivation of the Hilbert-Schmidt theorem.
- solve non-linear and singular integral equations.
- relate the integral equations with Green's function.

SYLLABUS OF GE-2 (ii)

Unit – 1 **(12 hours)**

Types of integral equations, Introduction and relation with linear differential equation, Volterra integral equations and its solutions, Method of resolvent kernels, Method of successive approximations, Decomposition method, Convolution type of equation, Method of Laplace transform.

Unit – 2 **(8 hours)**

System of Volterra integral equations, Solutions of Integro-differential equation, Abel's integral equation and its generalizations, Non-linear integral equations.

Unit – 3 **(12 hours)**

Fredholm integral equations and its solutions, Method of resolvent kernels, Method of successive approximations, Integral equations with degenerate kernels, Solutions of Fredholm integral equations using characteristic numbers and eigenfunctions with their properties.

Unit – 4 **(13 hours)**

Hilbert-Schmidt theorem, Non-homogeneous Fredholm integral equation with symmetric kernel, Fredholm alternatives, Applications of Green's function for solution of the boundary value problems, Singular integral equations, Applications of integral equations: Volterra's population growth model.

Tutorial

Problem-solving sessions based on material covered in the lectures will be taken up in the tutorials along with scholastic work related to conceptual understanding of the subject.

Essential Readings

[1] W. Hackbusch, *Integral Equations: Theory and Numerical Treatment*, Birkhäuser, 1995.
[2] M. L. Krasnov, A. I. Kiselev & G. I. Makarenko, *Problems and Exercises in Integral Equations*, Mir Publication Moscow, 1971.
[3] A. C. Pipkin, *A Course on Integral Equations*, Springer, 1991.
[4] A. M. Wazwaz, *Linear and Non-linear Integral Equations*, Springer 2011.

Suggested Readings

(i) S. G. Georgiev, *Integral Equations on Time Scales*, Atlantis Press, 2016.
(ii) F. B. Hildebrand, *Methods of Applied Mathematics*, Dover Publications, 1992.
(iii) J. D. Logan, *Applied Mathematics*, Fourth Edition, John Wiley & Sons, 2013.

Semester-III

Discipline Specific Core (DSC) Courses

DISCIPLINE SPECIFIC CORE – 7: FLUID DYNAMICS

CREDIT DISTRIBUTION OF THE COURSE

Course Title & Code	Credits	Credit Distribution of the Course			Eligibility Criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical		
DSC-7: Fluid Dynamics	4	3	1	0	Same as for entry to M.Sc. Mathematics	Knowledge of Calculus and basic Partial Differential Equations

Learning Objectives

The objective of this course is to:

- prepare a mathematical foundation to study the motion of fluids.
- develop concept, models, and techniques that enable to solve the problems of fluid flow.
- develop the ability to conduct advanced studies and research in the broad field of fluid dynamics.

Learning Outcomes

After studying this course, the student will be able to:

- understand the concept of fluids, their classification, flow lines, models and approaches to study fluid flow.
- formulate mass and momentum conservation principles and obtain their solution for non- viscous flow.
- know potential flow, Bernoulli's equation, Kelvin's minimum energy and circulation theorems.
- understand two- and three-dimensional motion, complex potential, circle theorem, Blasius theorem, Weiss's and Butler's sphere theorems.
- apply the concept of stress and strain in viscous flow to derive Navier–Stokes equation of motion and energy equation.

SYLLABUS OF DSC-7

Unit – 1 **(10 hours)**

Classification of fluids, Continuum model, Eulerian and Lagrangian approach of description, Differentiation following the fluid motion, Flow lines, vorticity and circulation, Conservation of mass: Equation of Continuity, Boundary surface.

Unit – 2 **(12 hours)**

Forces in fluid motion, Conservation of momentum: Euler's equation of motion, Theory of irrotational motion: Integration of Euler's equation under different conditions, Bernoulli's equation, Impulsive motion, Kelvin's minimum energy and circulation theorems, Potential theorem.

Unit – 3 **(13 hours)**

Two-dimensional motion: Complex potential, Line sources, sinks, doublets and vortices, Two-dimensional image system, Milne–Thomson circle theorem, Images with respect to a plane and cylinder, Blasius theorem. Three-dimensional flows, Weiss's sphere theorem, Images with respect to sphere, Axi-symmetric flow, Stokes stream function, Butler's sphere theorem, Flow past spheres and cylinders.

Unit – 4 **(10 hours)**

Stress and strain analysis, Newton's law of viscosity, Laminar flow, Navier-Stokes equation of motion, Steady flow between parallel planes and Poiseuille flow, Constitutive equation, Energy equation.

Tutorial

Problem-solving sessions based on material covered in the lectures will be taken up in the tutorials along with scholastic work related to conceptual understanding of the subject.

Essential Readings

[1] F. Chorlton, *Text Book of Fluid Dynamics*, CBS Publisher, 2005.
[2] R. W. Fox, P. J. Pritchard & A. T. McDonald, *Introduction to Fluid Mechanics*, Seventh Edition, John Wiley & Sons, 2009.
[3] P. K. Kundu, I. M. Cohen & D. R. Dowling, *Fluid Mechanics*, Sixth Edition, Academic Press, 2016.

Suggested Readings

(i) L. M. Milne-Thomson, *Theoretical Hydrodynamics*, The Macmillan company, USA, 1969.
(ii) D. E. Rutherford, *Fluid Dynamics*, Oliver and Boyd Ltd., 1978.

DISCIPLINE SPECIFIC CORE – 8: MEASURE AND INTEGRATION

CREDIT DISTRIBUTION OF THE COURSE

Course Title & Code	Credits	Credit Distribution of the Course			Eligibility Criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical		
DSC-8: Measure and Integration	4	3	1	0	Same as for entry to M.Sc. Mathematics	Knowledge of Real Analysis and Riemann Integration

Learning Objectives

The primary objective of this course is to:

- extend the notion of length of an interval with the introduction of the concept of Lebesgue outer measure for any subset of real line.
- investigate the properties of Lebesgue measurable sets and functions.
- familiarize students with the Lebesgue integration of functions and its comparison with Riemann integration.
- generalize the concepts of measure and integration to an abstract space.

Learning Outcomes

This course will enable the students to:

- verify whether a given subset of $\mathbb{R}$ or a real valued function is measurable.
- understand the requirement and the concept of the Lebesgue integral (a generalization of the Riemann integration) along with its properties.
- understand the statements and proofs of the fundamental integral convergence theorems and demonstrate their applications.
- carry out a comprehensive study of functions of bounded variation and their utility in understanding differentiation and integration.
- apply Hölder and Minkowski inequalities in L^p-spaces and understand completeness of L^p-spaces.

SYLLABUS OF DSC-8

Unit – 1 **(14 hours)**

Lebesgue outer measure, Measurable sets, Lebesgue measure, Borel sets, Regularity, Measurable functions, Borel and Lebesgue measurability, Non-measurable sets.

Unit – 2 **(13 hours)**

Integration of nonnegative functions, General integral, Integration of series, Riemann and Lebesgue integrals.

Unit – 3 **(8 hours)**

Functions of bounded variation, Lebesgue's differentiation theorem, Differentiation and integration, Absolute continuity of functions.

Unit – 4 **(10 hours)**

Measures and outer measures, Measure spaces, Integration with respect to a measure, L^p-spaces, Hölder's and Minkowski's inequalities, Completeness of L^p-spaces.

Tutorial

Problem-solving sessions based on material covered in the lectures will be taken up in the tutorials along with scholastic work related to conceptual understanding of the subject.

Essential Readings

[1] G. de Barra, *Measure Theory and Integration*, Ellis Horwood Ltd., Chichester, John Wiley & Sons, Inc., New York, 1981 (Indian Reprint, 2014).

Suggested Readings

(i) M. Capinski & P. E. Kopp, *Measure, Integral and Probability*, Springer, 2005.
(ii) E. Hewitt & K. Stromberg, *Real and Abstract Analysis: A Modern Treatment of the Theory of Functions of a Real Variable,* Springer, Berlin, 1975.
(iii) H. L. Royden & P.M. Fitzpatrick, *Real Analysis*, Fourth Edition, Pearson, 2015.

Semester-IV

Discipline Specific Core (DSC) Courses

DISCIPLINE SPECIFIC CORE – 9: PARTIAL DIFFERENTIAL EQUATIONS

CREDIT DISTRIBUTION OF THE COURSE

Course Title & Code	Credits	Credit Distribution of the Course			Eligibility Criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical		
DSC-9: Partial Differential Equations	4	3	1	0	Same as for entry to M.Sc. Mathematics	Knowledge of Multivariate Calculus and Differential Equations

Learning Objectives

The main objective of this course is to introduce:

- well-posedness, fundamental solutions, existence and uniqueness of solutions for Laplace equation, Poisson equation and Heat equation.
- solution for wave equation by spherical means.
- characteristics, complete integrals, envelopes and conservation laws for first-order nonlinear partial differential equations.
- classical solution techniques such as Green's function, similarity solutions and transform methods.

Learning Outcomes

This course will enable the students to:

- understand Laplace equation, Poisson equation, and Heat equation, their fundamental solutions, uniqueness principles, mean value properties, and Green's function.
- apply the method of spherical means to solve homogeneous and nonhomogeneous wave equations.
- use characteristics to solve nonlinear partial differential equations, construct complete integrals and envelopes, and understand conservation laws.
- implement various techniques such as similarity solutions and transform methods to derive solutions of different types of partial differential equations.

SYLLABUS OF DSC-8

Unit – 1 **(12 hours)**

Well-posed problems, Classical solution, Laplace equation, Poisson equation, Fundamental solution, Strong maximum principle and uniqueness of solution, Mean value formulas, Representation formula, Green's function, Poisson's formula.

Unit – 2 **(10 hours)**

Heat equation, Fundamental solution for homogeneous and nonhomogeneous initial-value problems, Mean value formula, Strong maximum principle and uniqueness of solution, Local estimates for the solution.

Unit – 3 **(13 hours)**

Wave equation: Solution of homogeneous and nonhomogeneous problems by spherical means, Nonlinear first order partial differential equations: Complete integrals and envelopes, Characteristics, Introduction to conservation laws.

Unit – 4 **(10 hours)**

Other solution methods: Similarity solutions, Fourier transform and Laplace transform, Cole-Hopf transformation, Potential function, Hodograph and Legendre transform.

Tutorial

Problem-solving sessions based on material covered in the lectures will be taken up in the tutorials along with scholastic work related to conceptual understanding of the subject.

Essential Readings

[1] L. C. Evans, *Partial Differential Equations*, American Mathematical Society, Providence, RI, 1998.
[2] F. John, *Partial Differential Equations*, Fourth Edition, Springer-Verlag, New York, 1982.

Suggested Readings

(i) P. R. Garabedian, *Partial Differential Equations*, John Wiley & Sons, Inc., New York- London-Sydney, 1964.
(ii) A. K. Nandakumaran & P. S. Datti, *Partial Differential Equations: Classical Theory with a Modern Touch*, Cambridge University Press, 2020.

DISCIPLINE SPECIFIC CORE – 10: ANALYSIS OF SEVERAL VARIABLES

CREDIT DISTRIBUTION OF THE COURSE

Course Title & Code	Credits	Credit Distribution of the Course			Eligibility Criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical		
DSC-10: Analysis of Several Variables	**4**	**3**	**1**	**0**	**Same as for entry to M.Sc. Mathematics**	**Knowledge of Calculus, Real Analysis including Riemann Integration**

Learning Objectives

The primary objective of this course is to:

- introduce differentiation of vector valued functions on $\mathbb{R}^n$ and their properties.
- familiarize students with integration of functions over rectangles and bounded sets in $\mathbb{R}^n$.
- extend integration of functions to unbounded sets in $\mathbb{R}^n$.
- study change of variables and its applications.

Learning Outcomes

This course will enable the students to:

- check differentiability of vector valued functions on $\mathbb{R}^n$, understand the relation between directional derivative and differentiability, apply chain rule, mean value theorems, inverse and implicit function theorems.
- understand higher order derivatives and be able to apply Taylor's formulas with integral remainder, Lagrange's remainder and Peano's remainder.
- master the concepts of integration over rectangles and bounded sets in $\mathbb{R}^n$.
- generalize the integration theory to unbounded sets in $\mathbb{R}^n$.
- grasp the effect of change of variables in integration.

SYLLABUS OF DSC-10

Unit– 1 **(12 hours)**

The differentiability of functions from $\mathbb{R}^n$ to $\mathbb{R}^m$, Partial derivatives and differentiability, Directional derivatives and differentiability, Chain rule, Mean value theorems, Inverse function theorem and Implicit function theorem.

Unit– 2 **(11 hours)**

Derivatives of higher order, Taylor's formulas with integral remainder, Lagrange's remainder and Peano's remainder, The integral over a rectangle, Existence of the integral.

Unit– 3 **(10 hours)**

Evaluation of the integral, Fubini's theorem, The integral over a bounded set.

Unit– 4 **(12 hours)**

Rectifiable sets, Improper integrals**,** The change of variable theorem, Applications of change of variables.

Tutorial

Problem-solving sessions based on material covered in the lectures will be taken up in the tutorials along with scholastic work related to conceptual understanding of the subject.

Essential Readings

[1] M. Giaquinta & G. Modica, *Mathematical Analysis: An Introduction to Functions of Several Variables*, Birkhäuser, 2009.
[2] J. R. Munkres, *Analysis on Manifolds*, CRC Press, Taylor & Francis, 2018.

Suggested Readings

(i) W. Rudin, *Principles of Mathematical Analysis*, Third Edition, Mc Graw Hill, 1986.
(ii) M. Spivak, *Calculus on Manifolds: A Modern Approach to Classical Theorems of Advanced Calculus*, Taylor & Francis, 2018.