Appendix-36
AC dated 12.07.2024

Summary of Changes in Programmes under UGCF

Programmes under UGCF-NEP
4.0.1.20(d) Renaming of Courses having Similar Content & Proposal for Additional Electives

As recommended by the OSD (Examinations) and discussed in the house, we have renamed Courses having Similar Content. New electives have been proposed based on suggestions from colleagues in the colleges.

Annexure-4.01.20(d)
Item No. 4-1-20
AC dated 12.07.2024

Renaming of Courses having Similar Content & Proposal for Additional Electives

The following table reflects the changes wrt courses (Semester I to Semester VI). The table also includes some of the courses that are already approved for semester I-Semester VI, and are also being proposed for semester VII and VIII and that will have identical names and syllabi and some of the newly proposed electives:

SNo	Programme	Semester	Existing	Proposed
1	B.Sc(H) Computer Science	I	DSC01: Programming using Python	DSC01: Object Oriented Programming using Python
2	B.A. Programme	I	DSC-A1: Programming Fundamentals using Python	DSC-A1: Programming using Python
3	Generic Elective	I	GE1a: Programming with Python	GE1a: Programming using Python
4	B.Sc(H) Computer Science	II	DSC04: Object Oriented Programming with C++	DSC04: Programming using C++
5	B.A. Programme	I	DSC-1: Introduction to Programming using C++	DSC01: Programming using C++
6	B.Sc. Programme	I	Programming Fundamentals using C++	DSC01: Programming using C++
7	Generic Elective	I	GE1b: Programming using C++	GE1b: Programming using C++
8	B.Sc(H) Computer Science	I	DSC02: Computer System Architecture	DSC02: Computer System Architecture
9	B.A. Programme	III	DSC03: Computer System Architecture	DSC03: Computer System Architecture
10	B.Sc. Programme	III	DSC03: Computer System Architecture	DSC03: Computer System Architecture
11	Generic Elective	II	N.A.	GE2c: Computer System Architecture
12	B.Sc(H) Computer	III	DSC07: Data Structures	DSC07: Data Structures

	Science			
13	B.A. Programme	II	DSC02: Data Structures	DSC02: Data Structures
14	B.Sc Programme	II	DSC02: Data Structures using C++	DSC02: Data Structures
15	Generic Elective	IV	GE4a: Data Structures using C++	GE4a: Data Structures
16	B.A. Programme	II	DSC-A2: Data Analysis and Visualization using Python	DSC-A2: Data Analysis and Visualization using Python
17	Generic Elective	II	GE2a: Data Analysis and Visualization using Python	GE2a: Data Analysis and Visualization using Python
18	Discipline Specific Elective	III	DSE: Data Analysis and Visualization (DAV)	DSE: Data Analysis and Visualization using Python
19	B.Sc(H) Computer Science	IV	DSC11: Database Management Systems	DSC11: Database Management Systems
20	B.A. Programme	V	DSC05: Database Management System	DSC05: Database Management Systems
21	B.Sc. Programme	V	DSC05: Database Management System	DSC05: Database Management Systems
22	Generic Elective	V	GE3a: Database Management Systems	GE3a: Database Management Systems
23	B.Sc(H) Computer Science	III	DSC08: Operating Systems	DSC08: Operating Systems
24	B.A. Programme	IV	DSC04: Operating Systems	DSC04: Operating Systems
25	B.Sc. Programme	IV	DSC04: Operating Systems	DSC04: Operating Systems
26	Generic Elective	V	GE5a: Operating Systems	GE5a: Operating Systems
27	B.Sc(H) Computer Science	IV	DSC12: Computer Networks	DSC12: Computer Networks
28	B.A. Programme	VI	DSC06: Computer Networks	DSC06: Computer Networks

29	B.Sc. Programme	VI	DSC06: Computer Networks	DSC06: Computer Networks
30	Generic Elective	VI	Sem VI: GE6a: Computer Networks	Sem VI: GE6a: Computer Networks
31	B.Sc(H) Computer Science	IV	DSC10: Design and Analysis of Algorithms	DSC10: Design and Analysis of Algorithms
32	B.A. Programme	VII	DSC07: Design and Analysis of Algorithms	DSC07: Design and Analysis of Algorithms
33	B.Sc. Programme	VII	DSC07: Design and Analysis of Algorithms	DSC07: Design and Analysis of Algorithms
34	Generic Elective	VII	GE7a: Design and Analysis of Algorithms	GE7a: Design and Analysis of Algorithms
35	B.A. Programme	III	DSC-A3: Data Mining-I	DSC-A3: Data Mining-I
36	Discipline Specific Elective	IV	DSE: Data Mining I	DSE: Data Mining-I
37	B.A. Programme	IV	DSC-A4: Data Mining-II	DSC-A4: Data Mining-II
38	Discipline Specific Elective	V	DSE: Data Mining-II	DSE: Data Mining-II
39	B.A. Programme	IV	GE4b: Introduction to Web Programming	GE4b: Introduction to Web Programming
40	Discipline Specific Elective	V	DSE: Introduction to Web Programming	DSE: Introduction to Web Programming
41	Generic Elective	VI	N.A.	GE6d: Data Privacy
42	Discipline Specific Elective	V	DSE: Data Privacy	DSE: Data Privacy
43	B.Sc(H) Computer Science	VI	DSC17: Machine Learning	DSC17: Machine Learning
44	B.A. Programme	V	DSC-A5: Machine Learning	DSC-A5: Machine Learning

45	Generic Elective	VII	GE7c: Machine Learning	GE7c: Machine Learning
46	B.Sc(H) Computer Science	VI	DSC16: Artificial Intelligence	DSC16: Artificial Intelligence
47	Generic Elective	VI	GE6c: Artificial Intelligence	GE6c: Artificial Intelligence
48	Discipline Specific Elective	V	N.A.	DSE: Artificial Intelligence
49	B.A. Programme	VII	DSC-A6: Deep Learning	DSC-A6: Deep Learning
50	Discipline Specific Elective	VI	DSE: Deep Learning	DSE: Deep Learning
51	Discipline Specific Elective	VI	N.A.	DSE: Numerical Optimization
52	Generic Elective	VII	N.A.	GE7e: Ethical Hacking
53	Discipline Specific Elective	VI	DSE: Ethical Hacking	DSE: Ethical Hacking
54	Generic Elective	VIII	N.A.	GE8d: Cyber Forensics
55	Discipline Specific Elective	VII	N.A.	DSE: Cyber Forensics
56	B.Sc(H) Computer Science	VIII	DSC20: Information Security	DSC20: Information Security
57	B.A. Programme	VIII	DSC08: Information Security	DSC08: Information Security
58	B.Sc. Programme	VIII	DSC08: Information Security	DSC08: Information Security
59	Generic Elective	VIII	GE8a: Information Security	GE8a: Information Security
60	Generic Elective	VIII	GE8c: Introduction to Parallel Programming	GE8c: Introduction to Parallel Programming
61	Discipline Specific Elective	VI	N.A.	DSE: Introduction to Parallel Programming

62	B.Sc(H) Computer Science	VI	DSC18: Cloud Computing	DSC18: Cloud Computing
63	Generic Elective	VII	N.A.	GE7d: Cloud Computing
64	Discipline Specific Elective	VIII	N.A.	DSE8e: Cloud Computing

Note: N.A. in the fourth column in the above table indicates a newly proposed course.

Syllabus

DSC01: OBJECT ORIENTED PROGRAMMING USING PYTHON

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		
Object Oriented Programming using Python	4	3	0	1	Class XII pass	Nil

Course Objectives:

This course introduces programming to a novice. Python is used for problem solving. The course also introduces the concept of object- oriented programming.

Learning Outcomes:

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

• Develop, document, and debug modular Python programs.

• Handle files.

• Apply suitable programming constructs and built-in data structures to solve a problem.

• Use classes and objects in application programs

Syllabus

Theory

Unit 1 Introduction to Programming: Problem solving strategies; Structure of a Python program; Syntax and semantics; Executing simple programs in Python.

Unit 2 Creating Python Programs: Identifiers and keywords; Literals, numbers, and strings; Operators; Expressions; Input/output statements; Defining functions; Control structures (conditional statements, loop control statements, break, continue and pass, exit function), default arguments.

Unit 3 Built-in data structures: Mutable and immutable objects; Strings, built-in functions for string, string traversal, string operators and operations; Lists creation, traversal, slicing and splitting operations, passing list to a function; Tuples, sets, dictionaries and their operations.

Unit 4 Object Oriented Programming: Introduction to classes, objects and methods; Standard libraries.

Unit 5 File and exception handling: File handling through libraries; Errors and exception handling.

Practical **(30 hours)**

List of Practicals

1. Write a program to find the roots of a quadratic equation
2. Write a program to accept a number 'n' and
   a. Check if 'n' is prime
   b. Generate all prime numbers till 'n'
   c. Generate first 'n' prime numbers This program may be done using functions
3. Write a program to create a pyramid of the character '*' and a reverse pyramid
4. Write a program that accepts a character and performs the following:
   a. print whether the character is a letter or numeric digit or a special character.
   b. if the character is a letter, print whether the letter is uppercase or lowercase
   c. if the character is a numeric digit, prints its name in text (e.g., if input is 9, output is NINE)
5. Write a program to perform the following operations on a string

a. Find the frequency of a character in a string.
b. Replace a character by another character in a string.
c. Remove the first occurrence of a character from a string.
d. Remove all occurrences of a character from a string.

6. Write a program to swap the first n characters of two strings.
7. Write a function that accepts two strings and returns the indices of all the occurrences of the second string in the first string as a list. If the second string is not present in the first string then it should return -1.
8. Write a program to create a list of the cubes of only the even integers appearing in the input list (may have elements of other types also) using the following:

    a. 'for' loop
    b. list comprehension

9. Write a program to read a file and
    a. Print the total number of characters, words and lines in the file.
    b. Calculate the frequency of each character in the file. Use a variable of dictionary type to maintain the count.
    c. Print the words in reverse order.
    d. Copy even lines of the file to a file named 'File1' and odd lines to another file named 'File2'.
10. Write a program to define a class Point with coordinates x and y as attributes. Create relevant methods and print the objects. Also define a method distance to calculate the distance between any two point objects.
11. Write a function that prints a dictionary where the keys are numbers between 1 and 5 and the values are cubes of the keys.
12. Consider a tuple t1=(1, 2, 5, 7, 9, 2, 4, 6, 8, 10). Write a program to perform following operations:
    a. Print half the values of the tuple in one line and the other half in the next line.
    b. Print another tuple whose values are even numbers in the given tuple.
    c. Concatenate a tuple t2=(11,13,15) with t1.
    d. Return maximum and minimum value from this tuple
13. Write a program to accept a name from a user. Raise and handle appropriate exception(s) if the text entered by the user contains digits and/or special characters.

Essential Readings

- Taneja, S., Kumar, N. Python Programming- A modular Approach, 1st edition, Pearson Education India, 2018.

- Balaguruswamy E. Introduction to Computing and Problem Solving using Python, 2nd edition, McGraw Hill Education, 2018.

Suggestive Readings

- Brown, Martin C. Python: The Complete Reference, 2nd edition, McGraw Hill Education, 2018.
- Guttag, J.V. Introduction to computation and programming using Python, 2nd edition, MIT Press, 2016.

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

DSC-A1/GE1a: PROGRAMMING USING PYTHON

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		
Programming using Python	4	3	0	1	Class XII pass	Nil

Course Objectives

This course is designed to introduce programming concepts using Python to students. The course aims to develop structured as well as object-oriented programming skills using Python. The course also aims to achieve competence amongst its students to develop correct and efficient Python programs to solve problems spanning multiple disciplines.

Learning Outcomes

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

- Write simple programs using built-in data types of Python.
- Implement arrays and user defined functions in Python.
- Solve problems spanning multiple disciplines using suitable programming constructs in Python.

Syllabus

Unit 1

Introduction to Python Programming: Problem solving strategies; Structure of a Python program; Syntax and semantics; Python interpreter/shell, indentation; Executing simple programs in Python.

Unit 2

Creating Python Programs: Identifiers and keywords; literals, numbers, and strings; Operators and expressions; Input and output statements; control structures (conditional statements, loop control statements, break, continue and pass), Errors and exception handling.

Unit 3

User Defined Functions: Defining functions, passing arguments and returning values, default arguments

Unit 4

Built-in data structures: Strings, Lists, Tuples, Sets, Dictionaries; their built-in functions, operators and operations.

References

1. Kamthane, A. N., & Kamthane, A.A. *Programming and Problem Solving with Python*, McGraw Hill Education. 2017.

2. Balaguruswamy E., "*Introduction to Computing and Problem Solving using Python*",2nd Edition, McGraw Hill Education, 2018.

3. Taneja, S., Kumar, N. *Python Programming- A modular Approach*. Pearson Education India, 2018.

Additional References

(i) Guttag, J. V. *Introduction to computation and programming using Python*. MIT Press. 2018

(ii) Downey, A. B. *Think Python–How to think like a Computer Scientist* 2nd Edition. O'Reilly 2015

Suggested Practical List

1. Write a program to calculate total marks, percentage and grade of a student. Marks obtained in each of three subjects are to be input by the user. Assign grades according to the following criteria:

   Grade A : if Percentage >=80
   Grade B : if Percentage >=60 and Percentage <80
   Grade C : if Percentage >=40 and Percentage <60
   Grade D : if Percentage <=40

2. Write a program to print factors of a given number.

3. Write a program to add N natural numbers and display their sum.

4. Write a program to print the following conversion table (use looping constructs):

Height(in Feet)	Height(in inches)
5.0ft	60 inches
5.1ft	61.2inches

.
.
.

5.8ft	69.6inches
5.9ft	70.8inches
6.0ft	72inches

5. Write a program that takes a positive integer n and the produce n lines of output as shown:

*

* *

* * *

* * * *

(for n =4)

6. Write a menu driven program using user defined functions to print the area of rectangle, square, circle and triangle by accepting suitable input from user.

7. Write a function that calculates factorial of a number n.

8. Write a program to print the series and its sum: (use functions)

   1/1! + 1/2! + 1/3! …….1/n!

9. Write a program to perform the following operations on an input string

   a. Print length of the string

   b. Find frequency of a character in the string

   c. Print whether characters are in uppercase or lowercase

10. Write a program to create two lists: one of even numbers and another of odd numbers. The program should demonstrate the various operations and methods on lists.

11. Write a program to create a dictionary where keys are numbers between 1 and 5 and the values are the cubes of the keys.

12. Write a program to create a tuple t1 = (1,2,5,7,2,4). The program should perform the following:

   a. Print tuple in two lines, line 1 containing the first half of tuple and second line having the second half.

   b. Concatenate tuple t2 = (10,11) with t1.

DSC04/DSC01/GE1b: PROGRAMMING USING C++

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		
Programming using C++	4	3	0	1	Class XII pass	NIL

Course Objectives:

This course is designed to introduce programming concepts using C++ to students. The course aims to develop structured as well as object-oriented programming skills using C++ programming language. The course also aims to achieve competence amongst its students to develop correct and efficient C++ programs to solve problems spanning multiple domains.

Learning outcomes

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

- Write simple programs using built-in data types of C++.
- Implement arrays and user defined functions in C++.
- Write programs using dynamic memory allocation, handling external files, interrupts and exceptions.
- Solve problems spanning multiple domains using suitable programming constructs in C++.
- Solve problems spanning multiple domains using oriented programming concepts in C++.

Syllabus

Unit-1 **(3 hours)**

Introduction to C++: Overview of Procedural and Object-Oriented Programming, Using main() function, Header Files, Compiling and Executing Simple Programs in C++.

Unit-2 **(12 hours)**

Programming Fundamentals: Data types, Variables, Operators, Expressions, Arrays, Keywords, Decision-making constructs, Iteration, Type Casting, Input-output statements, Functions, Command Line Arguments/Parameters

Unit-3 **(15 hours)**

Object Oriented Programming: Concepts of Abstraction, Encapsulation. Creating Classes and objects, Modifiers and Access Control, Constructors, Destructors, Implementation of Inheritance and Polymorphism, Template functions and classes

Unit-4 **(9 hours)**

Pointers and References: Static and dynamic memory allocation, Pointer and Reference Variables, Implementing Runtime polymorphism using pointers and references

Unit-5 **(6 hours)**

Exception and File Handling: Using try, catch, throw, throws and finally; Nested try, creating user defined exceptions, File I/O Basics, File Operations

Practical **(30 hours)**

1. Write a program to compute the sum of the first n terms of the following series: The number of terms n is to be taken from the user through the command line. If the command line argument is not found then prompt the user to enter the value of n.
2. Write a program to remove the duplicates from an array.
3. Write a program that prints a table indicating the number of occurrences of each alphabet in the text entered as command line arguments.
4. Write a menu driven program to perform string manipulation (without using inbuilt string functions):
   a. Show address of each character in string
   b. Concatenate two strings.
   c. Compare two strings
   d. Calculate length of the string (use pointers)
   e. Convert all lowercase characters to uppercase
   f. Reverse the string
   g. Insert a string in another string at a user specified position
5. Write a program to merge two ordered arrays to get a single ordered array.

6. Write a program to search a given element in a set of N numbers using Binary search
   a. with recursion
   b. without recursion.
7. Write a program to calculate GCD of two numbers
   a. with recursion
   b. without recursion.
8. Create a Matrix class. Write a menu-driven program to perform following Matrix operations (exceptions should be thrown by the functions if matrices passed to them are incompatible and handled by the main() function):
   a. Sum
   b. Product
   c. Transpose
9. Define a class Person having name as a data member. Inherit two classes Student and Employee from Person. Student has additional attributes as course, marks and year and Employee has department and salary. Write display() method in all the three classes to display the corresponding attributes. Provide the necessary methods to show runtime polymorphism.
10. Create a Triangle class. Add exception handling statements to ensure the following conditions: all sides are greater than 0 and sum of any two sides are greater than the third side. The class should also have overloaded functions for calculating the area of a right angled triangle as well as using Heron's formula to calculate the area of any type of triangle.
11. Create a class Student containing fields for Roll No., Name, Class, Year and Total Marks. Write a program to store 5 objects of Student class in a file. Retrieve these records from the file and display them.
12. Copy the contents of one text file to another file, after removing all whitespaces.

Essential/recommended readings

1. Stephen Prata, C++ Primer Plus, 6th Edition, Pearson India, 2015.
2. E Balaguruswamy, Object Oriented Programming with C++, 8th edition, McGraw- Hill Education, 2020. 3. D.S. Malik, C++ Programming: From Problem Analysis to Program Design, 6th edition, Cengage Learning, 2013.

Suggestive Readings

1. Schildt, H. C++: The Complete Reference, 4th edition, McGraw Hill, 2003

2. Forouzan, A. B., Gilberg, R. F. Computer Science: A Structured Approach using C++, 2nd edition, Cengage Learning, 2010

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

DSC02/DSC03/GE2c: COMPUTER SYSTEM ARCHITECTURE

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		
Computer System Architecture	4	3	0	1	Class XII pass	NIL

Course Objectives

The objectives of this course are as follows:

• Introduces the students to the fundamental concepts of digital computer organization, design and architecture.
• Develop a basic understanding of the building blocks of the computer system and highlight how these blocks are organized together to architect a digital computer system.

Learning Outcomes

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

• Design Combinational Circuits using basic building blocks. Simplify these circuits using Boolean algebra and Karnaugh maps. Differentiate between combinational circuits and sequential circuits.
• Represent data in binary form, convert numeric data between different number systems and perform arithmetic operations in binary.
• Determine various stages of instruction cycle, pipelining and describe interrupts and their handling.
• Describe how CPU communicates with memory and I/O devices.

• Distinguish between different types of processors.
• Simulate the design of a basic computer using a software tool.

Syllabus

Theory

Unit – 1 **(6 hours)**

Digital Logic Circuits

Logic Gates, Truth Tables, Boolean Algebra, Digital Circuits, Combinational Circuits, Introduction to Sequential Circuits, Circuit Simplification using Karnaugh Map, Don't Care Conditions, Flip-Flops, Characteristic Tables, Excitation Table.

Unit – 2 **(9 hours)**

Digital Components (Fundamental building blocks)

Designing of combinational circuits- Half Adder, Full Adder, Decoders, Encoders, Multiplexers, Registers and Memory (RAM, ROM and their types), Arithmetic Microoperations, Binary Adder, Binary Adder-Subtractor.

Unit – 3 **(6 hours)**

Data Representation and Basic Computer Arithmetic

Number System, r and (r-1)'s Complements, data representation and arithmetic operations.

Unit – 4 **(9 hours)**

Basic Computer Organization and Design

Bus organization, Microprogrammed vs Hardwired Control, Instruction Codes, Instruction Format, Instruction Cycle, Instruction pipelining, Memory Reference, Register Reference and Input Output Instructions, Program Interrupt and Interrupt Cycle..

Unit – 5 **(6 hours)**

Processors

General register organization, Stack Organization, Addressing Modes, Overview of Reduced Instruction Set Computer (RISC), Complex Instruction Set Computer (CISC), Multicore processor and Graphics Processing Unit (GPU).

Unit – 6 **(9 hours)**

Memory and Input-Output Organization

Memory hierarchy (main, cache and auxiliary memory), Input-Output Interface, Modes of Transfer: Programmed I/O, Interrupt initiated I/O, Direct memory access.

Essential Readings

• David A. Patterson and John L. Hennessy. "Computer Organization and Design: The Hardware/Software interface", 5th edition, Elsevier, 2012.

• Mano, M. Computer System Architecture, 3rd edition, Pearson Education, 1993.

Suggestive Readings

• Mano, M. Digital Design, Pearson Education Asia, 1995.

• Null, L., & Lobur, J. The Essentials of Computer Organization and Architecture. 5th edition, (Reprint) Jones and Bartlett Learning, 2018.

• Stallings, W. Computer Organization and Architecture Designing for Performance 8th edition, Prentice Hall of India, 2010

Practicals (30 hours)

(Use Simulator – CPU Sim 3.6.9 or any higher version for the implementation)

1. Create a machine based on the following architecture
2. Create a Fetch routine of the instruction cycle.
3. Write an assembly program to simulate ADD operation on two user-entered numbers.
4. Write an assembly program to simulate SUBTRACT operation on two user-entered numbers.
5. Write an assembly program to simulate the following logical operations on two user-entered numbers.
   AND, OR, NOT, XOR, NOR, NAND
6. Write an assembly program for simulating following memory-reference instructions.
   ADD
   LDA
   STA

BUN
ISZ

7. Write an assembly language program to simulate the machine for following register reference instructions and determine the contents of AC, E, PC, AR and IR registers in decimal after the execution:
   CLA
   CMA
   CME
   HLT
8. Write an assembly language program to simulate the machine for following register reference instructions and determine the contents of AC, E, PC, AR and IR registers in decimal after the execution
   INC
   SPA
   SNA
   SZE

9. Write an assembly language program to simulate the machine for following register reference instructions and determine the contents of AC, E, PC, AR and IR registers in decimal after the execution:
   CIR
   CIL \
10. Write an assembly program that reads in integers and adds them together; until a negative non-zero number is read in. Then it outputs the sum (not including the last number).
11. Write an assembly program that reads in integers and adds them together; until zero is read in. Then it outputs the sum.

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

DSC07/DSC02/GE4a: DATA STRUCTURES

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

	Credits	Credit distribution of the course		

Course title & Code		Lecture	Tutorial	Practical /Practice	Eligibility criteria	Pre-requisite of the course (if any)
Data Structures	**4**	**3**	**0**	**1**	**Passed 12th class with Mathematics**	Programming using C++

Course Objectives

The course aims at developing the ability to use basic data structures like arrays, stacks, queues, lists, and trees to solve problems. C++ is chosen as the language to implement the implementation of these data structures.

Learning Outcomes

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

● Compare two functions for their rates of growth.

● Understand abstract specification of data-structures and their implementation.

● Compute time and space complexity of operations on a data-structure.

● Identify the appropriate data structure(s) for a given application and understand the trade-offs involved in terms of time and space complexity.

● Apply recursive techniques to solve problems.

Syllabus

Unit-1 **(9 hours)**

Growth of Functions, Recurrence Relations: Functions used in analysis, asymptotic notations, asymptotic analysis, recurrence, Master Theorem.

Unit-2 **(16 hours)**

Arrays, Linked Lists, Stacks, Queues: Arrays: array operations, applications, two dimensional arrays, dynamic allocation of arrays; Linked Lists: singly linked lists, doubly linked lists, circularly linked lists, Stacks: stack as an ADT, implementing stacks using arrays, implementing stacks using

linked lists, applications of stacks; Queues: queue as an ADT, implementing queues using arrays, implementing queues using linked lists,. Time complexity analysis.

Unit-3 **(5 hours)**

Recursion: Recursive functions, linear recursion, binary recursion.

Unit-4 **(6 hours)**

Trees, Binary Trees: Trees: definition and properties, tree traversal algorithms, and their time complexity analysis; binary trees: definition and properties, traversal of binary trees, and their time complexity analysis.

Unit-5 **(7 hours)**

Binary Search Trees: Binary Search Trees: insert, delete, search operations, time complexity analysis of these operations

Unit-6 **(2 hours)**

Binary Heap: Binary Heaps: heaps, heap operations.

Essential/recommended readings

1. Goodrich, M.T., Tamassia, R., & Mount, D., Data Structures and Algorithms Analysis in C++, 2nd edition, Wiley, 2011. 4 th
2. Cormen, T.H., Leiserson, C.E., Rivest, R. L., Stein C. Introduction to Algorithms, edition, Prentice Hall of India, 2022. Additional references

Additional References

1. Sahni, S. Data Structures, Algorithms and applications in C++, 2nd edition, Universities Press, 2011.
2. Langsam Y., Augenstein, M. J., & Tanenbaum, A. M. Data Structures Using C and C++, Pearson, 2009.

Practical List **(30 hours)**

1. Write a program to implement singly linked list as an ADT that supports the following operations:
   a. Insert an element x at the beginning of the singly linked list
   b. Insert an element x at ith position in the singly linked list
   c. Remove an element from the beginning of the singly linked list
   d. Remove an element from ith position in the singly link

e. Search for an element x in the singly linked list and return its pointer

f. Concatenate two singly linked lists

2. Write a program to implement doubly linked list as an ADT that supports the following operations:
   a. Insert an element x at the beginning of the doubly linked list
   b. Insert an element x at ith position in the doubly linked list
   c. Insert an element x at the end of the doubly linked list
   d. Remove an element from the beginning of the doubly linked list
   e. Remove an element from ith position in the doubly linked list.
   f. Remove an element from the end of the doubly linked list
   g. Search for an element x in the doubly linked list and return its pointer (viii) Concatenate two doubly linked lists
3. Write a program to implement circular linked list as an ADT which supports the following operations:
   a. Insert an element x at the front of the circularly linked list
   b. Insert an element x after an element y in the circularly linked list
   c. Insert an element x at the back of the circularly linked list
   d. Remove an element from the back of the circularly linked list
   e. Remove an element from the front of the circularly linked list
   f. Remove the element x from the circularly linked list
   g. Search for an element x in the circularly linked list and return its pointer
   h. Concatenate two circularly linked lists
4. Implement a stack as an ADT using Arrays.
5. Implement a stack as an ADT using the Linked List ADT.
6. Write a program to evaluate a prefix/postfix expression using stacks.
7. Implement Queue as an ADT using the circular Arrays.
8. Implement Queue as an ADT using the Circular Linked List ADT.
9. Write a program to implement Binary Search Tree as an ADT having the following operations:
   a. Insert an element x
   b. Delete an element x
   c. Search for an element x in the BST and change its value to y and then place the node with value y at its appropriate position in the BST
   d. Display the elements of the BST in preorder, inorder, and postorder traversal
   e. Display the elements of the BST in level-by-level traversal
   f. Display the height of the BST

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

DSC-A2/GE2a/DSE: DATA ANALYSIS AND VISUALIZATION USING PYTHON

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
		Lecture	Tutorial	Practical/ Practice		
Data Analysis and Visualization using Python	**4**	**3**	**0**	**1**	**Class XII pass with Mathematics**	Programming using Python

Course Objectives

This course is designed to introduce the students to real-world data analysis problems, the use of statistics to get a deterministic view of data, and interpreting results in the field of exploratory data science using Python. This course is the first in the "Data Science" pathway and builds the foundation for three subsequent courses in the pathway.

Learning outcomes

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

1. Apply descriptive statistics to obtain a deterministic view of data

2. Perform data handling using Numpy arrays

3. Load, clean, transform, merge, and reshape data using Pandas

4. Visualize data using Pandas and matplot libraries

5. Solve real world data analysis problems

SYLLABUS OF DSE

Unit 1 (**10 hours)**

Introduction to basic statistics and analysis:

Fundamentals of Data Analysis, Statistical foundations for Data Analysis, Types of data, Descriptive Statistics, Correlation and covariance, Linear Regression, Statistical Hypothesis Generation and Testing, Python Libraries: NumPy, Pandas, Matplotlib, Seaborn

Unit 2 **(8 hours)**

Array manipulation using Numpy:

Numpy array: Creating Numpy arrays; various data types of Numpy arrays, indexing and slicing, swapping axes, transposing arrays, data processing using Numpy arrays.

Unit 3 **(12 hours)**

Data Manipulation using Pandas:

Data Structures in Pandas: Series, DataFrame, Index objects, Loading data into Pandas data frame, Working with DataFrames: Arithmetics, Statistics, Binning, Indexing, Filtering, Handling missing data, Hierarchical indexing, Data wrangling: Data cleaning, transforming, merging and reshaping

Unit 4 **(8 hours)**

Plotting and Visualization:

Using Matplotlib to plot data: figures, subplots, markings, color and line styles, labels and legends, Plotting functions in Pandas: Line, bar, Scatter plots, histograms, stacked bars, Heatmap, 3D Plotting, interactive plotting using Bokeh and Plotly.

Unit 5 **(7 hours)**

Data Aggregation and Group operations:

Group by mechanics, Data aggregation, General split-apply-combine, Pivot tables and cross tabulation

Essential/recommended readings

1. McKinney W. Python for Data Analysis: Data Wrangling with Pandas, NumPy and IPython, 2nd edition, O'Reilly Media, 2018.

2. Molin S. Hands-On Data Analysis with Pandas, Packt Publishing, 2019.

3. Gupta S.C., Kapoor V.K. Fundamentals of Mathematical Statistics,12 th edition, Sultan Chand & Sons, 2020.

Additional References

1. Chen D. Y. Pandas for Everyone: Python Data Analysis, First edition, Pearson Education, 2018.

2. Miller J.D. Statistics for Data Science, Packt Publishing Limited, 2017.

Suggested Practical List (If any): (30 Hours)

Practical exercises such as

Use a dataset of your choice from Open Data Portal (https:// data.gov.in/, UCI repository) or load from scikit, seaborn library for the following exercises to practice the concepts learnt.

1. Load a Pandas dataframe with a selected dataset. Identify and count the missing values in a dataframe. Clean the data after removing noise as follows

   a) Drop duplicate rows.
   b) Detect the outliers and remove the rows having outliers
   c) Identify the most correlated positively correlated attributes and negatively correlated attributes

2. Import iris data using sklearn library or (Download IRIS data from: https://archive.ics.uci.edu/ml/datasets/iris or import it from sklearn.datasets)

   i. Compute mean, mode, median, standard deviation, confidence interval and standard error for each feature
   ii. Compute correlation coefficients between each pair of features and plot heatmap
   iii. Find covariance between length of sepal and petal iv. Build contingency table for class feature
3. Load Titanic data from sklearn library , plot the following with proper legend and axis labels: a. Plot bar chart to show the frequency of survivors and non-survivors for male and female passengers separately

   b. Draw a scatter plot for any two selected features

c. Compare density distribution for features age and passenger fare
d. Use a pair plot to show pairwise bivariate distribution

4. Using Titanic dataset, do the following

   a. Find total number of passengers with age less than 30
   b. Find total fare paid by passengers of first class
   c. Compare number of survivors of each passenger class

5. Download any dataset and do the following

   a. Count number of categorical and numeric features
   b. Remove one correlated attribute (if any)

   c. Display five-number summary of each attribute and show it visually

Project: Students are encouraged to work on a good dataset in consultation with their faculty and apply the concepts learned in the course.

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

DSC11/DSC05/GE3a: DATABASE MANAGEMENT SYSTEMS

Credit distribution, Eligibility and Prerequisites 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 Systems	**4**	**3**	**0**	**1**	Pass in Class XII	NIL

Course Objectives

The course introduces the students to the fundamentals of database management system and its architecture. Emphasis is given on the popular relational database system including data

models and data manipulation. Students will learn about the importance of database structure and its designing using conceptual approach using Entity Relationship Model and formal approach using Normalization. The importance of file indexing and controlled execution of transactions will be taught. The course would give students hands-on practice of structured query language in a relational database management system and glimpse of basic database administration commands.

Learning outcomes

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

● Use database management system software to create and manipulate the database.

● Create conceptual data models using entity relationship diagrams for modeling real-life situations and designing the database schema.

● Use the concept of functional dependencies to remove redundancy and update anomalies.

● Apply normalization theory to get a normalized database scheme.

● Write queries using relational algebra, a procedural language.

Syllabus

Unit 1 **(5 hours)**

Introduction to Database: Purpose of database system, Characteristics of database approach, data models, database management system, database system architecture, three-schema architecture, components of DBMS, data independence, and file system approach vs database system approach.

Unit 2 **(7 hours)**

Entity Relationship Modeling: Conceptual data modeling - motivation, entities, entity types, attributes, relationships, relationship types, constraints on relationship, Entity Relationship diagram notation.

Unit 3 **(7 hours)**

Relational Data Model: Update anomalies, Relational Data Model - Concept of relations, schema-instance distinction, keys, relational integrity constraints, referential integrity and foreign keys, relational algebra operators and queries.

Unit 4 **(12 hours)**

Structured Query Language (SQL): Querying in SQL, DDL to create database and tables, table constraints, update database-update behaviors, DML, aggregation functions group by and having clauses, retrieve data from the database, generate and query views. Access and manipulate databases using ODBC. Basic Database administration SQL commands.

Unit 5 **(10 hours)**

Database Design: Mapping an Entity Relationship model to relational database, functional dependencies and Normal forms, 1NF, 2NF, 3NF and BCNF decompositions and desirable properties of them.

Unit 6 **(4 hours)**

Data Storage and Indexes: Need of file indexes, file organizations, index structures, single- and multi-level indexing, concurrent execution of transactions, ACID properties,.

Essential/recommended readings

1. Elmasri, R., Navathe, B. S. Fundamentals of Database Systems, 7th Edition, Pearson Education, 2015.

2. Krogh, J. W. MySQL Connector/Python Revealed: SQL and NoSQL Data Storage Using MySQL for Python Programmers, Apress, 2018.

3. Murach J. Murach's MySQL, 3rd edition, Pearson, 2019.

Additional References

1. Ramakrishnan, R., Gehrke J. Database Management Systems, 3rd Edition, McGraw Hill, 2014.

2. Silberschatz, A., Korth, H. F., Sudarshan S. Database System Concepts, 7th Edition, McGraw Hill, 2019.

3. Connolly, T. M., Begg, C. E. Database Systems: A Practical Approach to Design, Implementation, and Management, 6th edition, Pearson, 2019.

Practicals (30 hours)

Create and use the following student-course database schema for a college to answer the given queries using the standalone SQL editor.

Here, Rollno (ADMISSION) and SID (ADMISSION) are foreign keys. Note that course type may have two values viz. Fulltime and Parttime and a student may enroll in any number of courses

1. Retrieve names of students enrolled in any course.
2. Retrieve names of students enrolled in at least one part time course.
3. Retrieve students' names starting with letter 'A'.
4. Retrieve students' details studying in courses 'computer science' or 'chemistry'.
5. Retrieve students' names whose roll no either starts with 'X' or 'Z' and ends with '9'
6. Find course details with more than N students enrolled where N is to be input by the user.
7. Update student table for modifying a student name.
8. Find course names in which more than five students have enrolled
9. Find the name of youngest student enrolled in course 'BSc(P)CS'
10. Find the name of most popular society (on the basis of enrolled students)
11. Find the name of two popular part time courses (on the basis of enrolled students)
12. Find the student names who are admitted to full time courses only.
13. Find course names in which more than 30 students took admission
14. Find names of all students who took admission to any course and course names in which at least one student has enrolled
15. Find course names such that its teacher-in-charge has a name with 'Gupta' in it and the course is full time.
16. Find the course names in which the number of enrolled students is only 10% of its total seats.
17. Display the vacant seats for each course
18. Increment Total Seats of each course by 10%
19. Add enrollment fees paid ('yes'/'No') field in the enrollment table.
20. Update the date of admission for all the courses by 1 year.
21. Create a view to keep track of course names with the total number of students enrolled in it.
22. Count the number of courses with more than 5 students enrolled for each type of course.
23. Add column Mobile number in student table with default value '9999999999'
24. Find the total number of students whose age is > 18 years.
25. Find names of students who are born in 2001 and are admitted to at least one part time course.

Create and use the following student-society database schema for a college to answer the given (sample) queries using the standalone SQL editor.

II. Do the following database administration commands:

Create user, create role, grant privileges to a role, revoke privileges from a role, create index

DSC08/DSC04/GE5a: OPERATING SYSTEMS

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		
Operating Systems	4	3	0	1	Passed 12th class with Mathematics	Programming using C++ / Python/Java

Course Objectives

The course provides concepts that underlie all operating systems and are not tied to any particular operating system. The emphasis is on explaining the need and structure of an operating system using its common services such as process management (creation, termination etc.), CPU Scheduling, Process Synchronization, Handling Deadlocks, main memory management, virtual memory, secondary memory management. The course also introduces various scheduling algorithms, structures, and techniques used by operating systems to provide these services.

Learning Outcomes

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

● Describe the need of an operating system and define multiprogramming and Multithreading concepts.
● Implement the process synchronization service (Critical Section, Semaphores), CPU scheduling service with various algorithms.
● Implement Main memory Management (Paging, Segmentation) algorithms, Handling of Deadlocks
● Identify and appreciate the File systems Services, Disk Scheduling service

Syllabus

Unit-1 **(6 hours)**

Introduction: Operating Systems (OS) definition and its purpose, OS Structure, OS Operations: Dual and Multi-mode, OS as resource manager.

Unit-2 **(9 hours)**

Operating System Structures: OS Services, System Calls: Process Control, File Management, Device Management, and Information Maintenance, Inter-process Communication, and Protection, System programs, OS structure- Simple, Layered, Microkernel, and Modular.

Unit-3 **(10 hours)**

Process Management: Process Concept, States, Process Control Block, Process Scheduling, Schedulers, Context Switch, Operation on processes, Threads, Multicore Programming, Multithreading Models, Process Scheduling Algorithms: First Come First Served, Shortest-Job-First, Priority & Round-Robin, Process Synchronization: The critical section problem, Deadlock characterization, Deadlock handling.

Unit-4 **(11 hours)**

Memory Management: Physical and Logical address space, Swapping, Contiguous memory allocation strategies - fixed and variable partitions, Segmentation, Paging. Virtual Memory Management: Demand Paging and Page Replacement algorithms: FIFO Page Replacement, Optimal Page replacement, LRU page replacement.

Unit-5 **(9 hours)**

File System: File Concepts, File Attributes, File Access Methods, Directory Structure: Single Level, Two-Level, Tree-Structured, and Acyclic-Graph Directories. Mass Storage Structure: Magnetic Disks, Solid-State Disks, Magnetic Tapes, Disk Scheduling algorithms: FCFS, SSTF, SCAN, C-SCAN, LOOK, and C-LOOk Scheduling.

Essential/recommended readings

1. Silberschatz, A., Galvin, P. B., Gagne G. Operating System Concepts, 9 th edition, John Wiley Publications, 2016.

2. Tanenbaum, A. S. Modern Operating Systems, 3 rd edition, Pearson Education, 2007.
3. Stallings, W. Operating Systems: Internals and Design Principles, 9 th edition, Pearson Education, 2018.

Additional References

1. Dhamdhere, D. M., Operating Systems: A Concept-based Approach, 2nd edition, Tata McGraw-Hill Education, 2017.
2. Kernighan, B. W., Rob Pike, R. The Unix Programming Environment, Englewood Cliffs, NJ: Prentice-Hall, 1984.

Practicals

1. Execute various Linux commands for:
   a. Information Maintenance: wc, clear, cal, who, date, pwd
   b. File Management: cat, cp, rm, mv, cmp, comm, diff, find, grep, awk
   c. Directory Management : cd, mkdir, rmdir, ls
2. Execute various Linux commands for:
   a. Process Control: fork, getpid, ps, kill, sleep
   b. Communication: Input-output redirection, Pipe
   c. Protection Management: chmod, chown, chgrp
3. Write a programme (using fork() and/or exec() commands) where parent and child execute:
   a. same program, same code.
   b. same program, different code.
   c. Before terminating, the parent waits for the child to finish its task.
4. Write a program to to report behaviour of Linux kernel including kernel version, CPU type and model. (CPU information).
5. Write a program to report behaviour of Linux kernel including information on 19 configured memory, amount of free and used memory. (Memory information)
6. Write a program to copy files using system calls.
7. Use an operating system simulator to simulate operating system tasks.
8. Write a program to implement scheduling algorithms FCFS/ SJF/ SRTF/ non preemptive scheduling algorithms.
9. Write a program to calculate the sum of n numbers using Pthreads. A list of n numbers is divided into two smaller lists of equal size, and two separate threads are used to sum the sublists.
10. Write a program to implement first-fit, best-fit and worst-fit allocation strategies.

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

DSC12/DSC06/GE6a: COMPUTER NETWORKS

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		
Computer Networks	**4**	**3**	**0**	**1**	Pass in Class XII	NIL

Course Objectives

The course objectives of this paper are to:

• Understand the concepts behind computer networks and data communication.

• Learn the different types of networks, network topologies and their characteristics.

• Learn the working of protocols used at various layers.

• Understand the utility of different networking devices.

Learning Outcomes

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

• differentiate between various types of computer networks and their topologies.

• understand the difference between the OSI and TCP/IP protocol suit.

• distinguish between different types of network devices and their functions.

• design/implement data link and network layer protocols in a simulated networking environment.

Syllabus

Unit 1 **(8 hours)**

Introduction:

Types of computer networks, Internet, Intranet, network topologies (bus, star, ring, mesh, tree, hybrid topologies), network classifications. layered architecture approach, OSI Reference Model, TCP/IP Reference Model. Transmission Modes: simplex, half duplex and full duplex, network devices and their role.

Unit 2 **(9 hours)**

Physical Layer:

Analog signal, digital signal, the maximum data rate of a channel, transmission media (guided transmission media, wireless transmission, satellite communication), multiplexing (frequency division multiplexing, time-division multiplexing, wavelength division multiplexing). Guided Media (Wired) (Twisted pair, Coaxial Cable, Fiber Optics. Unguided Media (Radio Waves, Infrared, Micro-wave, Satellite).

Unit 3 **(10 hours)**

Data Link and MAC Layer:

Data link layer services, error detection and correction techniques, error recovery protocols (stop and wait, go back n, selective repeat), multiple access protocols with collision detection, MAC addressing, Ethernet..

Unit 4 **(8 hours)**

Network layer:

Networks and Internetworks, virtual circuits and datagrams, addressing, subnetting, Dijkstra Routing algorithm, Distance vector routing, Overview of Network Layer protocols- (ARP, IPV4, ICMP, RARP, IPV6)

Unit 5 **(10 hours)**

Transport and Application Layer:

Process to process Delivery- (client-server paradigm, connectionless versus connection-oriented service); User Datagram Protocols, TCP/IP protocol, Flow Control. FTP (File Transfer Protocol), SMTP (Simple Mail Transfer Protocol), Telnet (Remote login protocol), WWW (World Wide Web), HTTP (HyperText Transfer Protocol), URL (Uniform Resource Locator), DNS, DHCP, BOOTP.

Essential/recommended readings

1. Tanenbaum, A.S. & Wethrall, D.J.. Computer Networks, 5th edition, Pearson Education, 2012.

2. Forouzan, B. A.. Data Communication and Networking, 4th edition, McGraw-Hill Education, 2017.

Additional References

1. Comer, D. E.. Computer Networks and Internet, 6th edition, Pearson education, 2015.

2. Stallings, W., Data and Computer Communications, 10th edition, Pearson education India, 2017.

Practicals.

Introduce students to any network simulator tool and do the following:

1. To Study basic network command and Network configuration commands.

2. To study and perform PC to PC communication.

3. To create Star topology using Hub and Switch.

4. To create Bus, Ring, Tree, Hybrid, Mesh topologies.

5. Perform an initial Switch configuration.

6. Perform an initial Router configuration.

7. To implement Client Server Network.

8. To implement connection between devices using a router.

9. To perform remote desktop sharing within LAN connection.

DSC10/DSC07/GE7a: DESIGN AND ANALYSIS OF ALGORITHMS

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 and Analysis of Algorithms	**4**	**3**	**0**	**1**	**Pass in Class XII**	Data Structures

Course Objectives

The course is designed to develop understanding of different algorithm design techniques and use them for problem solving. The course shall also enable the students to verify correctness of algorithms and analyze their time complexity.

Learning Outcomes

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

- Compute and compare the asymptotic time complexity of algorithms.
- Use appropriate algorithm design technique(s) for solving a given problem.

Syllabus

Unit 1 **(8 hours)**

Searching, Sorting, Selection: Linear Search, Binary Search, Insertion Sort, Selection Sort, Bubble Sort, Heapsort, Linear Time Sorting, running time analysis and correctness.

Unit 2 **(5 hours)**

Graphs: Review of graph traversals, graph connectivity, testing bipartiteness, Directed Acyclic Graphs and Topological Ordering, Minimum Spanning Trees.

Unit 3 **(8 hours)**

Divide and Conquer: Introduction to divide and conquer technique, Merge Sort, Quick Sort, Randomised quicksort, Maximum-subarray problem, Strassen's algorithm for matrix multiplication.

Unit 4 **(5 hours)**

Greedy algorithms: Introduction to the Greedy algorithm design approach, application to minimum spanning trees, fractional knapsack problem, and analysis of time complexity.

Unit 5 **(5 hours)**

Dynamic Programming: Introduction to the Dynamic Programming approach, application to subset sum, integer knapsack problems, and analysis of time complexity.

Unit 6 **(4 hours)**

Hash Tables Hash Functions, Collision resolution schemes.

Essential/recommended readings

1. Cormen, T.H., Leiserson, C.E., Rivest, R. L., Stein C. Introduction to Algorithms, 4th edition, Prentice Hall of India, 2022.

2. Kleinberg, J., Tardos, E. Algorithm Design, 1st edition, Pearson, 2013.

Additional references

1. Basse, S., Gelder, A. V., Computer Algorithms: Introduction to Design and Analysis, 3rd edition, Pearson, 1999.

Practical List (If any): (30 Hours)

1. Write a program to sort the elements of an array using Insertion Sort (The program should report the number of comparisons).
2. Write a program to sort the elements of an array using Merge Sort (The program should report the number of comparisons).
3. Write a program to sort the elements of an array using Heap Sort (The program should report the number of comparisons).
4. Write a program to multiply two matrices using the Strassen's algorithm for matrix multiplication
5. Write a program to sort the elements of an array using Radix Sort.
6. Write a program to sort the elements of an array using Bucket Sort.

7. Display the data stored in a given graph using the Breadth-First Search algorithm.
8. Display the data stored in a given graph using the Depth-First Search algorithm.
9. Write a program to determine a minimum spanning tree of a graph using the Prim's algorithm.
10. Write a program to implement Dijkstra's algorithm to find the shortest paths from a given source node to all other nodes in a graph.
11. Write a program to solve the weighted interval scheduling problem.
12. Write a program to solve the 0-1 knapsack problem.

For the algorithms at S.No 1, 2 and 3 , test run the algorithm on 100 different input sizes varying from 30 to 1000. For each size find the number of comparisons averaged on 10 different input instances; plot a graph for the average number of comparisons against each input size. Compare it with a graph of nlogn.

DSC-A3/DSE: DATA MINING-I

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		
Data Mining - I	4	3	0	1	Passed 12th class with Mathematics	Programming using Python

Course Objectives

This course aims to introduce data mining techniques and their application on real-life datasets. The students will learn to pre-process the dataset and make it ready for application of data mining techniques. The course will focus on three main techniques of data mining i.e. Classification, Clustering and Association Rule Mining. Different algorithms for these techniques

will be discussed along with appropriate evaluation metrics to judge the performance of the results delivered.

Learning outcomes

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

● Pre-process the data for subsequent data mining tasks

● Apply a suitable classification algorithm to train the classifier and evaluate its performance.

● Apply appropriate clustering algorithm to cluster the data and evaluate clustering quality

● Use association rule mining algorithms and generate frequent item-sets and association rules

Syllabus

Unit 1 **(8 hours)**

Introduction to Data Mining:

Motivation and Challenges for data mining, Types of data mining tasks, Applications of data mining, Data measurements, Data quality, Supervised vs. unsupervised techniques

Unit 2 **(9 hours)**

Data Pre-Processing:

Data aggregation, sampling, dimensionality reduction, feature subset selection, feature creation, variable transformation.

Unit 3 **(11 hours)**

Cluster Analysis:

Basic concepts of clustering, measure of similarity, types of clusters and clustering methods, K-means algorithm, measures for cluster validation, determine optimal number of clusters

Unit 4 **(8 hours)**

Association Rule Mining:

Transaction data-set, frequent itemset, support measure, rule generation, confidence of association rule, Apriori algorithm, Apriori principle

Unit 5 **(9 hours)**

Classification:

Naive Bayes classifier, Nearest Neighbour classifier, decision tree, overfitting, confusion matrix, evaluation metrics and model evaluation.

Essential/recommended readings

1. Tan P.N., Steinbach M, Karpatne A. and Kumar V. Introduction to Data Mining,2nd edition, Pearson, 2021.

2. Han J., Kamber M. and Pei J. Data Mining: Concepts and Techniques, 3 rd edition, 2011, Morgan Kaufmann Publishers.

3. Zaki M. J. and Meira J. Jr. Data Mining and Machine Learning: Fundamental Concepts and Algorithms, 2nd edition, Cambridge University Press, 2020.

Additional References

1. Aggarwal C. C. Data Mining: The Textbook, Springer, 2015.
2. Dunham M. Data Mining: Introductory and Advanced Topics, 1st edition, Pearson Education India, 2006.

Recommended Datasets for :

Classification: Abalone, Artificial Characters, Breast Cancer Wisconsin (Diagnostic)

Clustering: Grammatical Facial Expressions, HTRU2, Perfume data

Association Rule Mining: MovieLens, Titanics

Practicals

1. Apply data cleaning techniques on any dataset (e,g, wine dataset). Techniques may include handling missing values, outliers, inconsistent values. A set of validation rules can be prepared based on the dataset and validations can be performed.
2. Apply data pre-processing techniques such as standardization/normalization, transformation, aggregation, discretization/binarization, sampling etc. on any dataset

3. Run Apriori algorithm to find frequent itemsets and association rules on 2 real datasets and use appropriate evaluation measures to compute correctness of obtained patterns a) Use minimum support as 50% and minimum confidence as 75% b) Use minimum support as 60% and minimum confidence as 60 % I.
4. Use Naive bayes, K-nearest, and Decision tree classification algorithms and build classifiers on any two datasets. Divide the data set into training and test set. Compare the accuracy of the different classifiers under the following situations: a) Training set = 75% Test set = 25% b) Training set = 66.6% (2/3rd of total), Test set = 33.3% II. Training set is chosen by i) hold out method ii) Random subsampling iii) Cross-Validation. Compare the accuracy of the classifiers obtained. Data is scaled to standard format.
5. Use Simple K-means algorithm for clustering on any dataset. Compare the performance of clusters by changing the parameters involved in the algorithm. Plot MSE computed after each iteration using a line plot for any set of parameters.

Project: Students should be promoted to take up one project on any UCI/kaggle/data.gov.in or a dataset verified by the teacher. Preprocessing steps and at least one data mining technique should be shown on the selected dataset. This will allow the students to have a practical knowledge of how to apply the various skills learnt in the subject for a single problem/project.

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

DSE-A4/DSE: DATA MINING-II

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		

Data Mining - II	**4**	**3**	**0**	**1**	Pass in Class XII	Data Mining-I

Course Objectives

The course introduces the students to the supervised and unsupervised learning techniques. Students will learn about the importance of ensemble methods, cluster analysis, anomaly detection and their applicability in mining patterns in real applications. At the end students will be exposed to two advanced topics: text mining and time-series mining. Students will use the learned topics in solving real applications using open-source software.

Learning outcomes

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

● Differentiate between partition-based, density-based and hierarchical clustering

● Build ensemble models to improve predictive performance of the classifier

● Identify anomalies and outliers using supervised and unsupervised techniques

● Analyze time-series data and extract patterns from the stamped data

● Mine textual data and do topic modelling

Syllabus

Unit 1 **(9 hours)**

Clustering:

Partitioning Methods, Hierarchical Methods, Density-Based Methods, Comparison of different methods

Unit 2 **(8 hours)**

Ensemble Methods:

Need of ensemble, Random Forests, Bagging and Boosting

Unit 3 **(10 hours)**

Anomaly Detection:

Outliers and Outlier Analysis, Outlier Detection Methods, Statistical Approaches, Proximity-based and density-based outlier detection, Clustering-based approaches

Unit 4 **(8 hours)**

Mining Text Data:

Document Preparation and Similarity, Clustering Methods for Text, Topic Modeling

Unit 5 **(10 hours)**

Stream Mining:

Time series basics, Date Ranges, Frequencies, and Shifting, Resampling and moving windows functions, Decay function, Clustering stamped data: STREAM and CluStream

Essential/recommended readings

1. Tan P.N., Steinbach M, Karpatne A. and Kumar V. Introduction to Data Mining, 2nd edition, Pearson, 2019.

2. Zaki M. J. and Meira J. Jr. Data Mining and Machine Learning: Fundamental Concepts and Algorithms, 2nd edition, Cambridge University Press, 2020.

3. Aggarwal C. C. Data Mining: The Textbook, Springer, 2015.

Additional References

1. Han J. Kamber M. and Pei J. Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, 2011.

2. Dunham M. Data Mining: Introductory and Advanced Topics, Pearson, 2006

Suggested Practicals List (If any): (30 Hours)

Practical exercise such as

1. Apply Partitioning Methods, Hierarchical Methods, Density-Based Methods for clustering on a data set and compare the performance of the obtained results using different metrics
2. Create an ensemble using Random Forest and show the impact of bagging and boosting on the performance
3. Apply different outlier-detection methods on a noisy dataset and compare their effectiveness in terms of outliers reported
4. Compute similarity between two documents after required document preparation
5. Considering a time-stamped data (sales data/weather data), compare the aggregate values visually using different moving windows function
6. Write a program to find the latent topics in a document using any topic modeling method and display top 5 terms that contribute to each topic along with their strength. Also, visualize the distribution of terms contributing to the topics.

Project: Students should be promoted to take up one project covering at least one unit of the syllabus on any UCI/kaggle/data.gov.in or a dataset verified by the teacher. This will allow the students to have a practical knowledge of how to apply the various skills learnt in the subject for a single problem/project.

GE4b/DSE: INTRODUCTION TO WEB PROGRAMMING

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		
Introduction to web programming	**4**	**3**	**0**	**1**	Pass in Class XII	NIL

Course Objectives

The course aims at introducing the basic concepts and techniques of client side web programming. The student shall be able to develop simple websites using HTML, CSS and Javascript.

Learning outcomes

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

● Build websites using the elements of HTML.

● Build dynamic websites using the client side programming techniques with CSS, Javascript and jQuery.

● Learn to validate client-side data

Syllabus

Unit 1 **(5 hours)**

Introduction:

Introduction to Internet and web design. Basic concepts of web architecture.

Unit 2 **(12 hours)**

HTML:

Introduction to hypertext mark-up language (html), creating web pages, lists, hyperlinks, tables, web forms, inserting images, frames.

Unit 3 **(8 hours)**

Cascading style sheet (CSS):

Concept of CSS, creating style sheet, Importing style sheets, CSS properties, CSS styling (background, text format, controlling fonts), CSS rules, Style Types, CSS Selectors, CSS cascade, working with block elements and objects, working with lists and tables, CSS id and class, box model (introduction, border properties, padding properties, margin properties).

Unit 4 **(10 hours)**

Javascript:

Document object model, data types and variables, functions, methods and events, controlling program flow, JavaScript object model, built-in objects and operators, validations.

Unit 5 **(10 hours)**

jQuery and JSON:

Introduction to jQuery, syntax, selectors, events. JSON file format for storing and transporting data.

Essential/recommended readings

1. Nixon, R. Learning PHP, MySQL & JavaScript with jQuery, CSS and HTML5, O'Rielly, 2018.

2. Powell, T.A. HTML & CSS: The Complete Reference, 5th edition, Tata McGrawHill, 2010.

3. Duckett, J. JavaScript and JQuery: Interactive Front-End Web Development, Wiley, 2014.

Additional References

1. Minnick, J. Web Design with HTML5 and CSS3, 8th edition, Cengage Learning, 2015.

2. Boehm, A., & Ruvalcaba, Z. Munarch's HTML5 and CCS, 4th edition, Mike Murach & Associates, 2018.

3. J. A. Ramalho Learn Advanced HTML 4.0 with DHTML, BPB Publications, 2007.

4. Ivan Bayross Web Enabled Commercial Application Development Using Html, Dhtml, Javascript, Perl CGI, BPB Publications, 2009.

Suggested Practical List (If any): (30 Hours)

Practical exercises such as

HTML

1. Create an HTML document with following formatting – Bold, Italics, Underline, Colors, Headings, Title, Font and Font Width, Background, Paragraph, Line Brakes, Horizontal Line, Blinking text as well as marquee text.
2. Create an HTML document with Ordered and Unordered lists, Inserting Images, Internal and External linking

3. Create an HTML displaying this semester's time table.
4. Create a website with horizontal and vertical frames. Top horizontal frame showing your college's name and logo. Bottom horizontal frame split into two vertical frames. The left frame with hyperlinks to pages related to faculty, courses, student activities, etc. The right frame showing corresponding pages based on the link clicked on the left frame.
5. Create a student registration form using HTML which has the following controls:

   Text Box
   Dropdown box
   Option/radio buttons

   Check boxes Reset and Submit button

CSS

Create a webpage for your department with drop down navigation menu for faculty, courses, activities, etc.. Implement the webpage using styles, rules, selectors, ID, class.

Javascript

1. Create event driven programs for the following:

   a. Enter a number and on click of a button print its multiplication table.
   b. Print the largest of three numbers entered by the user.
   c. Find the factorial of a number entered by the user.
   d. Enter a list of positive numbers using the prompt terminated by a zero. Find the sum and average of these numbers.

2. Create a student registration form using text, radio button, check box, drop down box, text field and all other required HTML elements. Customise the CSS and javascript to input and validate all data. Create functions to perform validation of each element, example:

   a. Roll number is a 7-digit numeric value
   b. Name should be an alphabetical value(String)

   c. Non-empty and valid fields like DOB

jQuery and JSON

1. Change text color and contents using button click events using jQuery

2. Select elements using ID, class, elements name, attribute name
3. Run code on click events in jQuery
4. Handle HTML form, store the data in JSON object, pass them to another page and display it there using jQuery/Javascript

GE6d/DSE: DATA PRIVACY

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
		Lecture	Tutorial	Practical/ Practice		
Data Privacy	**4**	**3**	**0**	**1**	**Pass in Class XII**	**NIL**

Objective:

This course aims to provides students with the ability to identify privacy related aspects of data uses, attacks on data privacy, evaluate proposed technical mechanisms for privacy protection and understand ethical issues related to data privacy

Course Learning Outcomes:

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

- Understand the basic principles of data privacy and the implications of data breaches.
- Identify and evaluate different methods of protecting sensitive data.

- Explain the role of privacy regulations in safeguarding personal information.
- Implement basic cryptographic techniques to secure data.
- Apply data anonymization techniques to protect personal information.
- Analyze the ethical considerations in data privacy.

Syllabus

Unit 1: Introduction to Data Privacy and Privacy Regulations

Definition of data privacy, Historical context of data privacy, Types of sensitive data, Privacy laws and regulations

Unit 2: Data Privacy Attacks, Cryptography and Data Protection

Type of Attacks/ Data Breaches on Data Privacy, Impact of Data Breaches / Attacks, Introduction to cryptography, Symmetric and asymmetric encryption, Hashing and digital signatures

Unit 3: Data Collection, Use and Reuse

Harms Associated with Data collections, use and reuse, Introduction to data anonymization, Data Anonymization Techniques for anonymizing data, Challenges in anonymizing data

Unit 4: Ethical considerations in Data Privacy

Privacy and Surveillance, Ethics of Data Collection and Use, Bias and discrimination in data analysis

References:

1. Ronald Leenes, Rosamunde van Brakel, and Serge Gutwirth: Data Protection and Privacy: The Age of Intelligent Machines, Hart Publishing, 2017.

2. Naavi: Personal Data Protection Act of India (PDPA 2020) : Be Aware, Be Ready and Be Compliant, 2020.

3. Ravinder Kumar Gaurav Goyal, The Right to Privacy in India: Concept and Evolution, Publisher: Lightning Source, 2016.

Additional References:

1. https://onlinecourses.nptel.ac.in/noc22_cs37/preview
2. https://www.coursera.org/learn/northeastern-data-privacy/home/info

Suggested Practicals:

Students may be asked to perform some of the following practical activities related to data privacy:

1. Data Privacy Audit: Students can conduct a data privacy audit of a company or organization to identify potential vulnerabilities and risks in their data privacy practices.

2. Privacy Impact Assessment: Students can conduct a privacy impact assessment (PIA) of a new technology or system to identify potential privacy risks and develop strategies to mitigate them.

3. Regulation Compliance: Students can explore the requirements of the Data Protection Regulations and develop a plan for ensuring compliance with the regulation.

4. Cryptography: Students can learn about different cryptographic techniques and tools, such as encryption, hashing, and digital signatures, and implement them in practice.

5. Anonymization Techniques: Students can learn about data anonymization techniques, such as k-anonymity, differential privacy, and data masking, and apply them to a real-world dataset.

6. Privacy Policy Analysis: Students can analyze the privacy policies of different companies and identify gaps or areas for improvement.

7. Privacy-Enhancing Technologies: Students can explore privacy-enhancing technologies (PETs), such as virtual private networks (VPNs), Tor, and secure messaging apps, and evaluate their effectiveness in protecting privacy.

8. Privacy Breach Response Plan: Students can develop a privacy breach response plan for a company or organization, including steps to take in the event of a data breach and strategies for communicating with affected parties.

9. Ethical Considerations: Students can explore ethical considerations in data privacy, such as the balance between privacy and security, the impact of data collection and analysis on marginalized communities, and the role of data ethics in technology development.

10. Case Studies: Students can analyze case studies of privacy breaches or successful privacy protection strategies, and identify key lessons and takeaways.

DSC17/DSC-A5/GE7c: MACHINE LEARNING

Credit distribution, Eligibility and Prerequisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course
		Lecture	Tutorial	Practical/ Practice		
Machine Learning	4	3	0	1	Pass in Class XII	Programming using Python/ bject Oriented Programming using Python

Course Objectives

The course aims at introducing the basic concepts and techniques of machine learning so that a student can apply machine learning techniques to a problem at hand.

Learning outcomes

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

• Differentiate between supervised and unsupervised learning tasks.

• State the need of preprocessing, feature scaling and feature selection.

• Formulate classification, regression and clustering problems as optimization problems

• Implement various machine learning algorithms learnt in the course.

SYLLABUS

Unit 1 **(5 Hours)**

Introduction:

Basic definitions and concepts, key elements, supervised and unsupervised learning, applications of ML.

Unit 2 **(8 Hours)**

Preprocessing:

Feature scaling, feature selection methods. dimensionality reduction (Principal Component Analysis), class balancing, outlier detection and removal.

Unit 3 **(12 Hours)**

Regression:

Linear regression with one variable, linear regression with multiple variables, gradient descent, over-fitting, regularization. Regression evaluation metrics.

Unit 4 **(12 Hours)**

Classification: Decision trees, Naive Bayes classifier, logistic regression, k-nearest neighbor classifier, perceptron, multilayer perceptron, neural networks, back-propagation algorithm, Support Vector Machine (SVM). Classification evaluation metrics

Unit 5 **(8 Hours)**

Clustering: Approaches for clustering, distance metrics, K-means clustering, hierarchical clustering.

Essential/recommended readings

1. Mitchell, T.M. Machine Learning, McGraw Hill Education, 2017.

2. James, G., Witten. D., Hastie. T., Tibshirani., R. An Introduction to Statistical Learning with Applications in R, Springer, 2014.

3. Alpaydin, E. Introduction to Machine Learning, MIT press, 2009.

Additional References

1. Flach, P., Machine Learning: The Art and Science of Algorithms that Make Sense of Data, Cambridge University Press, 2015.

2. Christopher & Bishop, M., Pattern Recognition and Machine Learning, New York: Springer-Verlag, 2016.

3. Sebastian Raschka, Python Machine Learning, Packt Publishing Ltd, 2019

Suggested Practical List:

Practical exercises such as

Use Python for practical labs for Machine Learning. Utilize publicly available datasets from repositories like https://data.gov.in/ and https://archive.ics.uci.edu/ml/datasets.php

For evaluation of the regression/classification models, perform experiments as follows:

• Scale/Normalize the data • Reduce dimension of the data with different feature selection techniques • Split datasets into training and test sets and evaluate the decision models • Perform k-cross-validation on datasets for evaluation

Report the efficacy of the machine learning models as follows: • MSE and R2 score for regression models • Accuracy, TP, TN, FP, TN, error, Recall, Specificity, F1-score, AUC for classification models

For relevant datasets make prediction models for the following

1. Naïve Bayes Classifier

2. Simple Linear Regression multiple linear regression

3. Polynomial Regression

4. Lasso and Ridge Regression

5. Logistic regression

6. Artificial Neural Network

7. k-NN classifier

8. Decision tree classification

9. SVM classification

10. K-Means Clustering

11. Hierarchical Clustering

DSC16/GE6e/DSE: ARTIFICIAL INTELLIGENCE

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
		Lecture	Tutorial	Practical/ Practice		
Artificial Intelligence	**4**	**3**	**0**	**1**	**Pass in Class XII**	**Programming using C++/Programming using Python/Object Oriented Programming using Python**

Course Objectives

The objectives of this course are to:

- To introduce basic concepts and techniques of Artificial Intelligence (AI).
- To apply informed search techniques for different applications.
- To learn various knowledge representation techniques and writing Prolog programs.
- To learn about the latest techniques for developing AI systems.

Learning outcomes

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

- identify problems that are amenable to solutions by specific AI methods.
- state the utility of different types of AI agents.
- apply different informed search techniques for solving problems.
- use knowledge representation techniques for AI systems.

SYLLABUS

Unit 1 **6 Hours**

Introduction: Introduction to artificial intelligence, background and applications, Turing test, Weak AI, Strong AI, Narrow AI, Artificial General Intelligence, Super AI, rational agent approaches to AI, introduction to intelligent agents, their structure, behavior and task environment.

Unit 2 **12 Hours**

Problem Solving and Searching Techniques: Problem characteristics, production systems, control strategies, breadth-first search, depth-first search, hill climbing and its variations, heuristics search techniques: best-first search, A* algorithm, constraint satisfaction problem, means-end analysis, introduction to game playing, min-max and alpha-beta pruning algorithms.

Unit 3 **16 Hours**

Knowledge Representation: Propositional logic, First-Order Predicate logic, resolution principle, unification, semantic nets, conceptual dependencies, frames, and scripts, production rules, Introduction to Programming in Logic (PROLOG).

Unit 4 **8 Hours**

Understanding Natural Languages: Components and steps of communication, the contrast between formal and natural languages in the context of grammar, Chomsky hierarchy of grammars, parsing, and semantics, Parsing Techniques, Context-Free and Transformational Grammars, Recursive transition nets.

Unit 5 **3 Hours**

AI The Present and the Future: Symbolic AI, Data-driven AI and Machine Learning, Introduction to Machine Learning and Deep Learning based AI, Interpretable and Explainable AI, Ethics of AI: benefits and risks of AI.

Essential/recommended readings

1. Russell, Stuart, J. and Norvig, Peter, *Artificial Intelligence - A Modern Approach*, Pearson, 4th edition, 2020..
2. Bratko, Ivan, *Prolog Programming for Artificial Intelligence*, Addison-Wesley, Pearson Education, 4th edition, 2012.
3. Patterson, DAN,W, *Introduction to A.I. and Expert Systems* – PHI, 2007.
4. Clocksin, W., F. and Mellish, *Programming in PROLOG*, 5th edition, Springer, 2003.

Additional references

1. Kaushik, Saroj, *Artificial Intelligence*, Cengage Learning India, 2011.
2. Rich, Elaine and Knight, Kelvin, *Artificial Intelligence*, 3rd edition, Tata McGraw Hill, 2010

Practical List :

Practical exercises such as

1. Write a program in Prolog to implement TowerOfHanoi(N) where N represents the number of disks.
2. Write a program to implement the Hill climbing search algorithm in Prolog.
3. Write a program to implement the Best first search algorithm in Prolog.
4. Write a program to implement A* search algorithm in Prolog.
5. Write a program to implement the min-max search algorithm in Prolog.
6. Write a program to solve the Water-Jug Problem in Prolog.
7. Implement sudoku problem (minimum 9×9 size) using constraint satisfaction in Prolog.
8. Write a Prolog program to implement the family tree and demonstrate the family relationship.
9. Write a Prolog program to implement knowledge representation using frames with appropriate examples.
10. Write a Prolog program to implement conc(L1, L2, L3) where L2 is the list to be appended with L1 to get the resulted list L3.
11. Write a Prolog program to implement reverse(L, R) where List L is original and List R is reversed list.
12. Write a Prolog program to generate a parse tree of a given sentence in English language assuming the grammar required for parsing.
13. Write a Prolog program to recognize context free grammar $a^n b^n$.

DSC-A6/DSE: DEEP LEARNING

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
		Lecture	Tutorial	Practical/ Practice		
Deep Learning	**4**	**3**	**0**	**1**	Pass in Class XII	Programming using Python/Object Oriented Programming using Python/Mathematics for Computing

Course Objectives

The objective of this course is to introduce students to deep learning algorithms and their applications in order to solve real problems.

Learning outcomes

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

• Describe the feed-forward and deep networks.

• Design single and multi-layer feed-forward deep networks and tune various hyper parameters.

• Implement deep neural networks to solve a problem

• Analyze performance of deep networks.

• Use pre-trained models to solve a problem.

SYLLABUS

Unit 1 **(8 Hours)**

Introduction to neural networks:

Artificial neurons, perceptron, computational models of neurons, Structure of neural networks, Multilayer feedforward neural networks (MLFFNN), Backpropagation learning, Empirical risk minimization, bias-variance tradeoff, Regularization, output units: linear, softmax , hidden units:tanh, RELU

Unit 2 **(8 Hours)**

Deep neural networks:

Difficulty of training DNNs, Greedy layerwise training, Optimization for training DNN's, Newer optimization methods for neural networks(AdaGrad, RMSProp, Adam), Regularization methods(dropout, drop connect, batch normalization).

Unit 3 **(8 Hours)**

Convolution neural networks(CNNs):

Introduction to CNN - convolution, pooling, Deep CNNs - LeNet, AlexNet. Training CNNs, weights initialization, batch normalization, hyperparameter optimization, Using a pre trained convnet

Unit 4 **(8 Hours)**

Recurrent neural networks (RNNs):

Sequence modeling using RNNs, Backpropagation through time, LongShort Term Memory (LSTM), Bidirectional RNN

Unit 5 **(8 Hours)**

Unsupervised deep learning:

Autoencoders, Generative Adversarial Networks.

Unit 6 **(5 Hours)**

Applications:

Computer vision, Speech recognition and NLP.

Essential/recommended readings

1. Ian Goodfellow, Yodhua Bengio and Aaron Courville, Deep Learning, MIT Press Book, 2016.

2. Francois Chollet, Deep Learning with python, 2nd edition, Meaning Publications Co, 2021.

Additional References

1. Bunduma, N., Fundamentals of Deep Learning, 1st edition, O'reilly Books, 2017.

2. Heaton, J., Deep Learning and Neural Networks, 1st edition, Heaton Research Inc., 2015.

Suggested Practical List :

Practical exercises such as

The following practicals are to be conducted using Python.

1. Implement a feed-forward neural networks for classifying movie reviews as positive or negative(using IMDB dataset)
2. Implement a deep-neural feed-forward network for estimating the price of house, given real-estate data(Boston Housing Price)
3. Implement a deep-neural network for classifying news wires by topic (Reuters dataset).
4. Implement CNN for classifying MNIST dataset
5. Create a model for time-series forecasting using RNN/LSTM 6. Implement an auto-encoder

DSE: NUMERICAL OPTIMIZATION

Credit distribution, Eligibility and Pre-requisites of the Course

Course title & Code	Credits	Credit distribution of the course	Eligibility criteria	

		Lecture	Tutorial	Practical/ Practice		Pre-requisite of the course (if any)
Numerical Optimization	**4**	**3**	**0**	**1**	**Passed 12th class with Mathematics**	**Programming using Python/ Programming using C++**

Course Objectives

The course aims to provide students with the experience of mathematically formulating a large variety of optimization/decision problems emerging out of various fields like data science, machine learning, business, and finance. The course focuses on learning techniques to optimize problems in order to obtain the best possible solution.

Learning outcomes

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

- Mathematically formulate the optimization problems using the required number of independent variables.
- Define constraint functions on a problem.
- Check the feasibility and optimality of a solution.
- Apply conjugate gradient method to solve the problem.

SYLLABUS

Unit 1 **(6 hours)**

Introduction: Mathematical Formulation using example, Continuous versus Discrete Optimization, Constrained and Unconstrained Optimization, Global and Local Optimization, Stochastic and Deterministic Optimization, Convexity, Optimization Algorithms

Unit 2 **(14 hours)**

Fundamentals of Unconstrained Optimization: Concept of a Solution - Recognizing a Local Minimum, Nonsmooth Problems, Overview of Algorithms

- Two Strategies: Line Search and Trust Region, Search Directions for Line Search Methods, Models for Trust-Region Methods, Scaling. Line Search - Convergence of Line Search Methods, Rate of Convergence - Convergence Rate of Steepest Descent; Newton's Method, Quasi-Newton Methods. Trust Region - The Cauchy Point Algorithm; Global Convergence - Reduction Obtained by the Cauchy Point; Convergence to Stationary Points.

Unit 3 **(7 hours)**

Conjugate Gradient Methods: Basic Properties of the Conjugate Gradient Method, A Practical Form of the Conjugate Gradient Method, and Rate of Convergence

Unit 4 **(8 hours)**

Calculating Derivatives: Finite-Difference Derivative Approximations, Approximating the Gradient, Approximating a Sparse Jacobian, Approximating the Hessian, Approximating a Sparse Hessian

Unit 5 **(10 hours)**

Theory of Constrained Optimization: Local and Global Solutions, Smoothness, Examples - A Single Equality Constraint, A Single Inequality Constraint, Two Inequality Constraints, Tangent Cone and Constraint Qualifications, First-Order Optimality Condition, Second-Order Conditions - Second-Order Conditions and Projected Hessians. Linear and non-linear constrained optimization, augmented Lagrangian Method

Essential/recommended readings

1. J. Nocedal and S.J. Wright, *Numerical Optimization*, 2nd edition, Springer Series in Operations Research, 2006.
2. A, Mehra, S Chandra, Jayadeva, *Numerical Optimization with Applications*, Narosa Publishing House, New Delhi, 2009,

Additional References

1. R. W. Hamming, *Numerical Methods for Scientists and Engineers*, 2nd edition, Dover Publications, 1986.

2. Q. Kong, T. Siauw, A. Bayen, *Python Programming and Numerical Methods: A Guide for Engineers and Scientists*, 1st edition, 2020.

Suggested Practical List (If any)

:(30 Hours)

Practical exercises such as

Write programs to implement the following methods:

Constrained and Unconstrained Optimization, Global and Local Optimization, Line Search and Trust Region, Convergence of Line Search Methods, Rate of Convergence - Convergence Rate of Steepest Descent, Newton's Method, Quasi-Newton Methods, The Cauchy Point algorithm, Finite-Difference Derivative Approximations, Convergence to Stationary Points, Conjugate Gradient Method, Rate of Convergence, Approximating a Sparse Jacobian, Approximating the Hessian, Approximating a Sparse Hessian, First-Order Optimality Condition, Second-Order Conditions - Second-Order Conditions, and Projected Hessians. Linear and non-linear constrained optimization Augmented Lagrangian Methods.

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

GE7e/DSE: ETHICAL HACKING

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
		Lecture	Tutorial	Practical/ Practice		

Ethical Hacking	**4**	**3**	**0**	**1**	**Pass in Class XII**	**NIL**

Course Objectives

The objective of this course is to enable students to be part of such a team that can conduct the security assessment of an organization through the process of ethical hacking. This course will introduce the students, the idea of security assessment of systems and networks under investigation and how to perform them under the legal and ethical framework. Further, this course will outline the importance of various stages of ethical hacking, including but not limited to tasks such as penetration testing, and usage of various tools at each stage.

Learning outcomes

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

1. Understand and acknowledge the relevance of legal, ethical, and professional challenges faced by an ethical hacker.
2. Apply fundamental principles of system, application, and network security to ethically attack / penetrate the system to uncover the security flaws.
3. Perform evaluation of security systems through a systematic ethical hacking process and recommend countermeasures to improve security.
4. Understand and use various tools and techniques used in various stages of the ethical hacking process.

Syllabus

Unit 1 **(4 Hours)**

Introduction: Overview of information security threats and attack vectors, vulnerability assessment and penetration testing concepts, information security controls, security laws and standards. OWASP top 10 vulnerabilities

Unit 2 **(6 hours)**

Footprinting and Reconnaissance: Introduction to network reconnaissance tools such ipconfig, ifconfig, domain tools, nmap, Wireshark, etc.

Unit 3 **(8 hours)**

Scanning and Enumeration: Network penetration testing, Password cracking techniques and countermeasures, NetBIOS tools

Unit 4 **(8 hours)**

Gaining and Maintaining Access: Network level attacks and countermeasures, Metasploit framework, Burp Suite

Unit 5 **(8 hours)**

Exploitation and Covering Tracks: Privilege escalation, social Engineering**,** identity theft, countermeasures, Covering tracks using attrib command and creating Alternate Data Stream (ADS) in Windows, Erasing evidence from Windows logs, Strategies for maintaining access.

Unit 6 **(8 hours)**

Advanced stages: Denial of service, Session hijacking, hacking web servers, hacking web applications, sql injection etc.

Unit 7 **(8 hours)**

NIST Cybersecurity framework and ISO standards: NIST cybersecurity framework, Cyber Kill chain, ISO/IEC 27001 and related standards.

Unit 8 **(4 Hours)**

Cyber Defense and Reporting: Preparing vulnerability assessment reports, presenting post testing findings, preparing recommendations

References

1. Patrick Engbretson, The Basics of Hacking and Penetration Testing, 2nd Edition, Syngress, 2013.
2. Georgia Weidman, Penetration TEsting: A Hands-On Introduction to Hacking, 1st Edition, No Starch Press, 2014.

Additional References

1. Peter Kim, The Hacker Playbook 3: Practical Guide to Penetration Testing, Zaccheus Entertainment, 2018.
2. Jon Erickson, Hacking: The Art of Exploitation, No Starch Press, 2008.
3. Online Resources:

https://www.sans.org/cyberaces/

https://skillsforall.com/

https://www.hackingloops.com/ethical-hacking/

Suggested Practical List (If any): (30 Hours)

Perform the following activities, record and report in standard form.

(NOTE: Exercise extra caution while performing these exercises and codes)

1. Perform various Virtual Machine based exercises on https://vulnhub.com/
2. Perform Capture the Flag (CTF) exercises from https://www.hacker101.com/
3. Follow the lessons and activities from https://www.hackingloops.com/ethical-hacking/
4. Google site for hacking https://google-gruyere.appspot.com/
5. OWASP WebGoat https://github.com/WebGoat/WebGoat

GE8d/DSE: CYBER FORENSICS

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course
		Lecture	Tutorial	Practical/ Practice		

Cyber Forensics	**4**	**3**	**0**	**1**	**Pass in Class XII**	**NIL**

Course Objective:

This course is to equip students with the knowledge and skills necessary to identify, collect, analyze and present digital evidence in a manner that is admissible in legal proceedings. Students will be able to conduct a thorough investigation of cybercrime incidents, preserve digital evidence, and report findings to relevant stakeholders.

Course Learning Outcomes:

- Students will be able to demonstrate an understanding of the principles of digital forensics, including legal considerations, recognition, collection, and preservation of digital evidence.
- Students will develop skills in using digital forensics tools and techniques, such as creating disk images, conducting keyword and grep searches, and examining Windows registry.
- Students will learn evidence recovery methods, including deleted file recovery, formatted partition recovery, and data recovery procedures, and ethical considerations.
- Students will gain knowledge of cyber forensic investigation tools and techniques, including digital evidence collection, preservation, and password cracking.
- Students will understand cyber laws and crimes, including hacking, viruses, intellectual property, and e-commerce, and the legal system of information technology, including jurisdiction issues and security and evidence in e-commerce.

Unit 1 – Digital Forensics: Introduction to digital forensics, legal considerations, recognising and collecting digital evidence, preservation of evidence, hash values and file hashing, creating disk images, keyword and grep searches, network basics, reporting and peer review, digital forensics report.

Unit 2 – Windows OS Forensics: Bits, bytes, Endieness, Disk partition schema, File systems – FAT, NTFS, ex-FAT, windows registry forensics, examining windows registry, NTUser.Dat Hive File Analysis, SAM Hive file, Software Hive file, System Hive File, USRClass.dat Hive File, AmCache Hive File.

Unit 3 – Evidence Recovery: Introduction to Deleted File Recovery, Formatted Partition Recovery, Data Recovery Tools, Data Recovery Procedures and Ethics, Complete time line analysis of computer files based on file creation, File modification and file access, Recover Internet Usage Data, Recover Swap Files/Temporary Files/Cache Files, Introduction to Encase Forensic Edition, Forensic Tool Kit (FTK), Use computer forensics software tools to cross validate findings in computer evidence.

Unit 4 – Investigation: Introduction to Cyber Forensic Investigation, Investigation Tools, Digital Evidence Collection, Evidence Preservation, E-Mail Investigation, E-Mail Tracking, IP Tracking, E-Mail Recovery, Encryption and Decryption methods, Search and Seizure of Computers, Recovering deleted evidences, Password Cracking.

Unit 5 – Cyber Crimes and Cyber Laws: Introduction to IT laws & Cyber Crimes, Internet, Hacking, Cracking, Viruses, Software Piracy, Intellectual property, Legal System of Information Technology, Understanding Cyber Crimes in context of Internet, Indian Penal Law & Cyber Crimes Fraud Hacking Mischief, International law, E-Commerce-Salient Features On-Line contracts Mail Box rule Privities of, Contracts Jurisdiction issues in E-Commerce Electronic Data Interchange, Security and Evidence in E-Commerce Dual Key encryption Digital signatures security issues.

References:

1. Marjee T. Britz, Computer Forensics and Cyber Crime: An Introduction, Pearson Education, 2013.

2. C. Altheide& H. Carvey Digital Forensics with Open Source Tools, Syngress, 2011. ISBN: 9781597495868.

Additional References:

1. Computer Forensics: Investigating Network Intrusions and Cybercrime" by Cameron H. Malin, Eoghan Casey, and James M. Aquilina

2. Online Course management System: https://esu.desire2learn.com/

3. Computer Forensics, Computer Crime Investigation by John R,Vacca, Firewall Media, New Delhi.

4. Computer Forensics and Investigations by Nelson, Phillips Enfinger, Steuart, CENGAGE Learning

5. Real Digital Forensics by Keith j.Jones, Richard Bejitlich,Curtis W.Rose ,AddisonWesley Pearson Education

Suggested Practicals

It is suggested that the following tools/e-resources can be explored during the practical sessions • Wireshark • COFEE Tool • Magnet RAM Capture • RAM Capture • NFI Defragger • Toolsley • Volatility

1. Study of Network Related Commands (Windows)
2. Study of Network related Commands(Linux)
3. Analysis of windows registry
4. Capture and analyze network packets using Wireshark. Analyze the packets captured.
5. Creating a Forensic image using FTK Imager/ Encase Imager: creating forensic image, check integrity of data, analyze forensic image
6. Using System internal tools for network tracking and process monitoring do the following:
    a. Monitor live processes
    b. Capture RAM
    c. Capture TCP/UDP packets
    d. Monitor Hard disk
    e. Monitor Virtual Memory
    f. Monitor Cache Memory

DSC20/DSC08/GE8a: INFORMATION SECURITY

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course
		Lecture	Tutorial	Practical/ Practice		

Information Security	**4**	**3**	**0**	**1**	**Pass in Class XII**	**NIL**

Course Objective

The goal of this course is to make a student learn basic principles of information security. Over the due course of time, the student will be familiarized with cryptography, authentication and access control methods along with software security. Potential security threats and vulnerabilities of systems are also discussed along with their impacts and countermeasures. This course also touches upon the implications of security in cloud and Internet of Things (IoT).

Learning Outcomes

On successful completion of this course, a student will be able to

- Identify the major types of threats to information security.
- Describe the role of cryptography in security.
- Discover the strengths and weaknesses of private and public key cryptosystems.
- Identify and apply various access control and authentication mechanisms.
- Discuss data and software security and related issues.
- Explain network security threats and attacks.
- Articulate the need for security in cloud and IoT.

Syllabus

Unit 1 **(3 hours)**

Overview: Computer Security Concepts, Threats, Attacks, and Assets, Security Functional Requirements, Fundamental Security Design Principles.

Unit 2 **(6 hours)**

Cryptographic tools: Confidentiality with Symmetric Encryption, Message Authentication and Hash Functions, Public-Key Encryption, Digital Signatures and Key Management, Random and Pseudorandom Numbers, Practical Application: Encryption of Stored Data.

Unit 3 **(10 hours)**

User authentication and Access Control: Digital User Authentication Principle, Password-Based Authentication, Remote User Authentication, Security Issues for User Authentication

Access Control Principles, Subjects, Objects, and Access Rights, Discretionary Access Control, Example: UNIX File Access Control, Role-Based Access Control, Attribute-Based Access Control, Identity, Credential, and Access Management, Trust Frameworks.

Unit 4 **(5 hours)**

Database and Data Center Security:

The Need for Database Security, SQL Injection Attacks, Database Access Control.

Unit 5 **(8 hours)**

Software Security: Types of Malicious Software, Advanced Persistent Threat, Propagation — Infected Content - Viruses, Propagation — Vulnerability Exploit - Worms, Propagation — Social Engineering — SPAM E-Mail, Trojans, Payload — System Corruption, Payload — Attack Agent — Zombie, Bots, Payload — Information Theft — Keyloggers, Phishing, Spyware, Payload — Stealthing — Backdoors, Rootkits, Countermeasures. **Overflow Attacks** - Stack Overflows, Defending Against Buffer Overflows, Other Forms of Overflow Attacks. **Software Security Issues** - Handling Program Input, Writing Safe Program Code, Handling Program Input.

Unit 6 **(6 hours)**

Network Security: Denial-of-Service Attacks, Flooding Attacks, Distributed Denial-of-Service Attacks, Overview of Intrusion Detection, Honeypots, The Need for Firewalls, Firewall Characteristics and Access Policy, Types of Firewalls, Public-Key Infrastructure.

Unit 7 **(7 hours)**

Wireless, Cloud and IoT Security: Cloud Computing, Cloud Security Concepts, Cloud Security Approaches, The Internet of Things, IoT Security. Wireless Security Overview, Mobile Device Security.

References

1. W. Stallings, L. Brown, *Computer Security: Principles and Practice*, 4th edition, Pearson Education, 2018.

Additional References

1. Pfleeger C.P., Pfleeger S.L., Margulies J. *Security in Computing*, 5th edition, Prentice Hall, 2015.
2. Lin S., Costello D.J., *Error Control Coding: Fundamentals and applications,* 2nd edition, Pearson Education, 2004.
3. Stallings W. *Cryptography and network security*, 7th edition, Pearson Education, 2018.
4. Berlekamp E. *Algebraic Coding Theory*, World Scientific Publishing Co., 2015.

5. Stallings W. *Network security essentials Applications and Standards*, 6th edition, Pearson Education, 2018.
6. Whitman M.E., Mattord H.J., *Principle of Information Security*, 6th edition, Cengage Learning, 2017.
7. Bishop M., *Computer Security: Art and Science*, 2nd Revised edition, Pearson Education, 2019.
8. Anderson R.J., *Security Engineering: A guide to building Dependable Distributed Systems*, 2nd edition, John Wiley & Sons, 2008.

Suggested Practical List

1. Demonstrate the use of Network tools: ping, ipconfig, ifconfig, tracert, arp, netstat, whois.
2. Use of Password cracking tools : John the Ripper, Ophcrack. Verify the strength of passwords using these tools.
3. Use nmap/zenmap to analyze a remote machine.
4. Use Burp proxy to capture and modify the message.
5. Implement caesar cipher substitution operation.
6. Implement monoalphabetic and polyalphabetic cipher substitution operation.
7. Implement playfair cipher substitution operation.
8. Implement hill cipher substitution operation.
9. Implement rail fence cipher transposition operation.
10. Implement row transposition cipher transposition operation.
11. Implement product cipher transposition operation.

GE8c/DSE: INTRODUCTION TO PARALLEL PROGRAMMING

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
		Lecture	Tutorial	Practical/ Practice		

Introduction to Parallel Programming	**4**	**3**	**0**	**1**	**Pass in Class XII**	**Computer System Architecture/A course in C++at class XII/Data Structures, Operating Systems**

Course Objective

The course introduces the students to the basic concepts and techniques of parallel programming. It enables them to design and implement parallel algorithms. The course would give the students hands-on practice to write parallel programs using shared and distributed memory models using OpenMP and Message Passing Interface (MPI).

Course Learning Outcomes

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

1. Appreciate the need of Parallel algorithms
2. Describe architectures for parallel and distributed systems.
3. Develop elementary parallel algorithms in shared memory models.
4. Develop elementary parallel algorithms in distributed memory models.

Syllabus

Unit 1

Introduction to Parallel Computing: Trends in microprocessor architectures, memory system performance, dichotomy of parallel computing platforms, physical organization of parallel platforms, communication costs in parallel machines, SIMD versus MIMD architectures, shared versus distributed memory, PRAM shared-memory model, distributed-memory model.

Unit 2

OpenMP programming for shared memory systems: Thread Basics, Controlling Thread and Synchronization Attributes, Multi-thread and multi-tasking, Context Switching, Basic OpenMP thread functions, Shared Memory Consistency Models and the Sequential Consistency Model, Race Conditions, Scoping variables, work-sharing constructs, critical sections, atomic operations, locks, OpenMP tasks, Introduction to tasks, Task queues and task execution, Accessing variables in tasks, Completion of tasks and scoping variables in tasks.

Unit 3

MPI programming for distributed memory systems: MPI basic communication routines (Introduction to MPI and basic calls, MPI calls to send and receive data, MPI call for broadcasting data, MPI Non-blocking calls, Introduction to MPI Collectives, Types of interconnects (Characterization of interconnects, Linear arrays, 2D mesh and torus, cliques)

Unit 4

Applications: Matrix-matrix multiply, Odd-Even sorting, distributed histogram, Breadth First search, Dijkstra's algorithm.

References

1. Grama, A., Gupta, A., Karypis, G., Kumar, V., *Introduction to Parallel Computing*, 2nd edition, Addison-Wesley, 2003.
2. Quinn, M., *Parallel Programming in C with MPI and OpenMP*, 1st Edition, McGraw-Hill, 2017.
3. Revdikar, L., Mittal, A., Sharma, A., Gupta, S., *A Naïve Breadth First Search Approach Incorporating Parallel Processing Technique For Optimal Network Traversal*, International Journal of Advanced Research in Computer and Communication Engineering Vol. 5, Issue 5, May 2016

Additional references

(i) B. Parhami, Introduction to Parallel Processing: Algorithms and Architectures, Plenum, 1999, Springer.

Suggested Practical List

1. Implement Matrix-Matrix Multiplication in parallel using OpenMP
2. Implement distributed histogram Sorting in parallel using OpenMP
3. Implement Breadth First Search in parallel using OpenMP
4. Implement Dijkstra's Algorithm in parallel using OpenMP

DSC17/GE7d/DSE8e: CLOUD COMPUTING

Credit distribution, Eligibility and Pre-requisites of the Course

	Credits	Credit distribution of the course		

Course title & Code		Lecture	Tutorial	Practical/ Practice	Eligibility criteria	Pre-requisite of the course
Cloud Computing	4	3	0	1	Pass in Class XII	NIL

Course Objective:

The objective of an undergraduate cloud computing course is to provide students with a comprehensive understanding of cloud computing technologies, services, and applications.

Course Learning Outcomes:

Learning outcomes for an undergraduate course on cloud computing may include:

1. Knowledge of the fundamental concepts and principles of cloud computing, including virtualization, scalability, reliability, and security.
2. Ability to design, develop, and deploy cloud-based applications using popular cloud platforms and services.
3. Familiarity with cloud computing architectures, including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS).
4. Visualize the economic, legal, and ethical implications of cloud computing, including issues related to data privacy, ownership, and security.
5. Ability to evaluate and select cloud-based solutions based on their technical, economic, and business requirements.
6. Understanding of the broader societal and environmental impacts of cloud-based services and applications.

Syllabus:

Unit 1: Overview of Computing Paradigm
Recent trends in Computing : Grid Computing, Cluster Computing, Distributed Computing, Utility Computing, Cloud Computing,

Unit 2: Introduction to Cloud Computing
Introduction to Cloud Computing, History of Cloud Computing, Cloud service providers, Benefits and limitations of Cloud Computing,

Unit 3: Cloud Computing Architecture

Comparison with traditional computing architecture (client/server), Services provided at various levels, Service Models- Infrastructure as a Service(IaaS), Platform as a Service(PaaS), Software as a Service(SaaS), How Cloud Computing Works, Deployment Models- Public cloud, Private cloud, Hybrid cloud, Community cloud, Case study of NIST architecture.

Unit 4: Case Studies

Case study of Service model using Google Cloud Platform (GCP), Amazon Web Services (AWS), Microsoft Azure, Eucalyptus.

Unit 5: Cloud Computing Management

Service Level Agreements(SLAs), Billing & Accounting, Comparing Scaling Hardware: Traditional vs. Cloud, Economics of scaling.

Unit 6: Cloud Computing Security

Infrastructure Security- Network level security, Host level security, Application level security, Data security and Storage- Data privacy and security Issues, Jurisdictional issues raised by Data location, Authentication in cloud computing.

References:

1. Thomas Erl, Ricardo Puttini and Zaigham Mahmood, Cloud Computing: Concepts, Technology and Architecture, Publisher: PHI, 2013.
2. Rajkumar Buyya, James Broberg, and Andrzej Goscinski, Cloud Computing: Principles and Paradigms, Wiley, 2013.
3. Boris Scholl, Trent Swanson, and Peter Jausovec, Cloud Native: Using Containers, Functions, and Data to Build Next-Generation Applications, Publisher : Shroff/O'Reilly, 2019.

Additional References:

1. *Cloud Computing Bible,* Barrie Sosinsky, *Wiley-India,* 2010
2. *Cloud Computing: Principles and Paradigms,* Editors: Rajkumar Buyya, James Broberg, Andrzej M. Goscinski, *Wile,* 2011

3. *Cloud Computing: Principles, Systems and Applications, Editors:* Nikos Antonopoulos, Lee Gillam, *Springer*, 2012

4. *Cloud Security: A Comprehensive Guide to Secure Cloud Computing*, Ronald L. Krutz, Russell Dean Vines, *Wiley-India*, 2010

Suggested Practical List:

1. Create virtual machines that access different programs on same platform.

2. Create virtual machines that access different programs on different platforms .

3. Working on tools used in cloud computing online-

    a) Storage

    b) Sharing of data

    c) manage your calendar, to-do lists,

    d) a document editing tool

4. Exploring Google cloud

5. Exploring microsoft cloud

6. Exploring amazon cloud