



दिल्ली विश्वविद्यालय University of Delhi

परिषद शाखा - II, कमरा नं: 211, नया प्रशासनिक खंड, दिल्ली विश्वविद्यालय, दिल्ली-110007, दूरभाष: 011-27001156/54
Council Branch - II, Room No. 211, New Administrative Block, University of Delhi, Delhi - 110007, Ph: 27001156/54

CNC-II/093/1/Misc./2026/50

Dated: 03.09.2026

NOTIFICATION

Sub: Amendment to Ordinance V

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

Add the following:

The following two Skill Enhancement Courses (SECs) are added to the existing pool of SECs based on Undergraduate Curriculum Framework 2022, and are hereby notified for the information of all concerned :

- (i) Introduction to C++
- (ii) Introduction to AI

Syllabus is at Annexure-1.

REGISTRAR

SEC 01: Introduction to C++

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Introduction to C++	2	1	0	1	Pass in Class XII	Nil

Course Objective

This course is designed to introduce C++ as a programming language. It provides an elementary understanding of programming through C++.

Course Learning Outcomes

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

- Create classes and reuse them in C++.
- Develop modular programs using C++.

Syllabus**Unit-1****(7 hours)**

Programming Fundamentals: Syntax of a program in C++, main() function, Header Files, Data types, Variables, Operators, Expressions, Arrays, Keywords, Decision-making constructs, Iteration, Type Casting, Input-output statements, Functions, Static and dynamic memory allocation, Pointer and Reference Variables. Compiling and Executing Simple Programs in C++.

Unit-2**(8 hours)**

Object Oriented Programming and Exception Handling: Creating Classes and objects, Modifiers and Access Control, Constructors, Destructors, Implementation of Inheritance and Polymorphism, Exception Handling: Using try, catch, throw, throws and finally

References

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

Additional References

- (i) Herbert Schildt, C++: *The Complete Reference*, 4th Edition, McGraw Hill, 2003

Suggested Practical list

All the programs must be written using functions where input is passed as a parameter and output is returned to the main function.

1. Write a program to compute the sum of the first n terms of the following series:

$$S = 1 - \frac{1}{2^2} + \frac{1}{3^3} - \frac{1}{4^4} + \frac{1}{5^5} - \dots$$

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. (i) with recursion (ii) without recursion.
7. Write a program to calculate GCD of two numbers (i) with recursion (ii) 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.
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.

NS

19/12/23

Revised syllabus with suggested

SEC02: Introduction to AI *Readings*

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 AI	2	1	0	1	Pass in Class XII	

Course Objectives

The objectives of this course are to:

1. Understand the foundations of modern generative AI across multiple modalities, including text, vision, speech, and images, and explain the basic principles behind how these systems work.
2. Build practical AI applications using APIs and open models for language generation, vision-language understanding, speech recognition, speech synthesis, and image generation.
3. Design multi-step AI workflows that combine prompting, memory, tool use, retrieval, and generation across modalities, and choose appropriate techniques for different real-world tasks.
4. Compare the capabilities, limitations, and responsible use considerations of different AI modalities and deployment settings.

Learning Outcomes

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

- Explain the core ideas behind large language models, vision-language models, speech AI systems, and image generation models.
 - Build practical AI applications using prompting, structured outputs, retrieval, external tools, and API-based workflows across text, image, and speech tasks.
 - Compare and select suitable approaches for language, multimodal, speech, and image-generation problems.
 - Evaluate the strengths, limitations, and responsible use concerns of modern generative AI systems in different modalities.
- nm*

SYLLABUS

Unit 1 (8 Hours)

Large Language Models (LLMs)

Introduction to artificial intelligence and modern generative AI, overview of large language models and their applications, differences between discriminative and generative models, APIs versus open-source models, limitations and opportunities of LLM systems.

Prompting with LLMs: system and user prompts, zero-shot and few-shot prompting, prompt templates, summarization, extraction, classification, question answering, and conversational assistance.

Structured generation and workflows: JSON/schema-constrained outputs, prompt chaining, reliability, common failure cases, and building simple LLM-powered applications using hosted APIs, building agentic systems with memory and tools

Vision-Language Models (VLMs)

Overview of multimodal AI and vision-language models, combining text and image understanding, applications in image captioning, visual question answering, document understanding, chart interpretation, and multimodal assistants.

Working with vision-language APIs: image inputs, multimodal prompting, asking grounded questions about images, extracting information from visual content, and comparing text-only and multimodal workflows.

Limitations, practical use cases, and design of simple applications using hosted vision-language APIs.

Unit 2 (7 Hours)

Speech AI: Speech-to-Text (STT) and Text-to-Speech (TTS)

Overview of speech AI, speech interfaces, and applications such as transcription, subtitle generation, voice assistants, accessibility tools, and spoken interaction systems.

Speech-to-text systems: audio input, transcription workflows, practical issues such as noise, accents, formatting, and post-processing of transcripts.

Text-to-speech systems: speech synthesis, voice options, style and language settings, and building simple speech-enabled applications using hosted STT and TTS APIs.

Image Generation

Overview of generative image AI and its applications in design, media, creativity, education, and communication.

Image generation workflows using APIs: text-to-image generation, prompt design, prompt refinement, style

control, image editing, and iterative generation.

Practical use cases, limitations, safety considerations, and development of simple applications using hosted image generation APIs.

Suggested Reading:

1. Alammar, J., & Grootendorst, M. (2025). An Illustrated Guide to AI Agents: Concepts and code for building agents with LLMs, tools, and memory. O'Reilly Media.
2. Alammar, J., & Grootendorst, M. (2024). Hands-on large language models: Language understanding and generation. O'Reilly Media.
3. OpenAI. (2026). API overview. OpenAI API Reference.
<https://developers.openai.com/api/reference/overview>

Suggested Practical List

Project: Students will propose, design, develop and deploy a practical application of their choosing.

1. Write a Python program using a hosted large language model API to build a text-based AI application such as summarization, extraction, question answering, classification, or conversational assistance.
2. Write a Python program using a hosted large language model API to produce structured output in a schema such as JSON for a practical information extraction or reporting task.
3. Build an agentic LLM which can use tools and store past context with memory.
4. Write a Python program using a hosted vision-language model API to build an application such as image captioning, visual question answering, document understanding, or chart interpretation.
5. Write a Python program using a hosted multimodal API to process both text and image inputs for a practical task such as grounded question answering or visual content analysis.
6. Write a Python program using a hosted speech-to-text API to build an application such as audio transcription, subtitle generation, spoken query processing, or meeting-note preparation.
7. Write a Python program using a hosted text-to-speech API to build an application such as speech synthesis for accessibility, announcements, reading assistance, or multilingual voice output.
8. Write a Python program using a hosted image generation API to build an application such as text-to-image generation, image editing, poster design, or creative content generation.
9. Write a Python program using one or more hosted AI APIs to build an integrated multimodal application combining text, image, and speech capabilities.

Nr
3/8/26