B.Sc.-I Semester (NEP-Revised Syllabus 2024)

Course Code:	P1	Course Title:	PROGRAMMING METHODOLOGY

Course Objectives:

- ✓ Aim to equip students with fundamental programming principles, problem-solving skills, and the ability to design and implement software solutions efficiently with C/C++.
- ✓ Learn about the strategies of writing efficient and well-structured computer algorithms/programs.

UNIT-	TOPICS
UNIT- I	Introduction to programming Concept of Problem Solving, General Problem Solving Algorithmic & Heuristics, Steps of problem solving. Problem definition, Program Concept, Characteristics of Programming, Program design, Stages in Program Development, Types of Programming Methodologies viz. top-down and bottom-up programming
Unit-II	Algorithms, Notations, Design, Pseudocode, Flowcharts, Flowchart symbols, making Flowchart using tool like Raptor tool. The Program Development- Development Cycle Coding & Documenting-Testing &debugging, Program Design-Modular Programming, Structured programming, Programming Conventions. Programming Languages Types, Examples, Translators -Assemblers, Interpreters & Compilers, Editors, Debuggers.
Unit-III	Basic programming concept Variables and constants, Input, Processing Data, Output Data; Data types Integer, Float, Character, Boolean, String etc.- Format of variable names, Type specific operations - arithmetic, logical, relational etc, Built-in functions for operation
Unit-IV	Predefined Functions, Programmer-defined Function, Local Variable, Function Overloading, Control flow statements, loops Functions with Default Arguments, Call-By-Value and Call-By-Reference Parameters, Recursion.
Suggested Readings (Text/Reference Books):	
1.	Problem Solving and Program Design in C, J. R. Hanly and E. B. Koffman, Pearson, 2015
2.	Programming and problem solving with C++, N. Dale and C. Weems, Jones & Bartlett

Course Code:	P2	Course Title:	FUNDAMENTALS OF COMPUTERS
Course Objectives:			
✓ . Provide students with a comprehensive understanding of computer Organisation, Including the basic components of a computer system. ✓ Cover the basics of weighted and non-weighted codes, memory storage units etc.			
UNIT-	TOPICS		
UNIT- I	Definition of Computer, Historical Evolution of computer, Computer System Concept, Characteristics, capabilities and limitations, Generation of computers, Classification of computer, Basic Components of computers: Control Unit, ALU, Memory, Input/Output function, Block diagram, Hardware (different types of hardware components), software Memory Hierarchy: Main memory. Auxiliary memory, Cache memory, RAM, ROM, SRAM, DRAM etc. Introduction to BIOS and UEFI, Booting Process, Stored Program Concept, Processor and its types, Motherboard and its component		
Unit-II	Peripherals, Input Devices- Keyboard, Mouse, Scanners, Bar-code Reader, Light pen, Justification Touch screen etc. Output Devices Monitor, Printers, Plotters, Sound card and Speakers, Storage Devices- Primary vs Secondary Storage, Magnetic Tape, Magnetic Disks, Cartridge tape, Data drives, Hard disk, Floppy disk, Optical disk, Pen drive		
Unit-III	Software (System Software, Application Software, E-accessibility Software) (Open source, freeware, and proprietary software), Computer Programming Language and its types.		
Unit-IV	Introduction to number system- Binary, Octal, Decimal, Hexadecimal (characteristics of each number system), Conversion from one number system to other, complements of number systems and arithmetic operations, Computer codes (BCD, EBCDIC, ASCII Code, Gray code, Excess-3 code and Unicode), Logic gates, Boolean algebra (rules, laws, De Morgan Theorem, Boolean expressions, and simplifications), Bit, BYTE, Nibble and larger units of memory		
Suggested Readings (Text/Reference Books):			
1.	Digital fundamentals Thomas Lloyd, PEARSON EDUCATION publication, Eleventh edition-Global Edition, ISBN10:1-292-07598-8, ISBN13:978-1-292-075983		
2.	. Digital Electronics-principles and integrated circuits. Anil K. Maini. Wiley publications,		
3.	Digital Computer Fundamentals, Thomas C Bartee, McGraw Hill Publisher, , 4thedition		
4.	. Introduction to Computer Science, ITL Education Solutions Pvt. Ltd., Pearson Education		

Course Code:	Q-1	Course Title:	Programming Concepts

Course Objectives:

- ✓ Aim to equip students with fundamental programming principles, problem-solving skills,
- ✓ Learn about the strategies of writing efficient and well-structured computer algorithms/programs.

UNIT	TOPICS
Unit-I	Introduction to programming- Concept of Problem Solving, General Problem Solving-Algorithmic & Heuristics, Steps of problem solving, Problem definition, Program Concept,
Unit-II	Characteristics of Programming, Program design, Stages in Program Development, Types of Programming Methodologies viz. top-down and bottom-up programming.
Unit-III	Algorithms, Notations, Design, Pseudocode, Flowcharts, Flowchart symbols, making Flowchart using tool like Raptor tool. The Program Development-Development Cycle Coding & Documenting-Testing & debugging, Program Design-Modular
Unit-IV	Programming, Structured programming, Programming Conventions. Programming Languages Types, Examples, Translators -Assemblers, Interpreters & Compilers, Editors, Debuggers

Suggested Readings (Text/Reference Books):

1. Problem Solving and Program Design in C. J. R. Hanly and E. B. Koffman, Pearson, 2015
2. Programming and problem solving with C++, N. Dale and C. Weems, Jones & Bartlett Learning, 2010

UNIVERSITY OF LUCKNOW
DEPARTMENT OF COMPUTER SCIENCE
Lucknow, Uttar Pradesh PIN-226007

B.Sc.-II Semester (NEP-Revised Syllabus 2024)

Course Code:	: P3	Course Title:	DATA STRUCTURES 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.			

UNIT	TOPICS
UNIT- I	Introduction to C++ Programming-Character Set, Keywords, Variables: declaration and initialization, constant and its types, data type, types of operators, operator precedence and associativity, type conversion and typecasting, Basic Program Structure in C++, Input and Output statements,
Unit-II	Selection Statements if, if else, nested if else, else if ladder, switch case and Iterative Statements for, while, do while, nesting of loops, break and continue statements, Functions, types of functions, passing values to functions, Recursion and recursive functions, Arrays notation and representation, single and multidimensional arrays
Unit-III	Asymptotic notations, asymptotic analysis, Time complexity analysis, solving recurrences using recursion trees. 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.
Unit-IV	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: Recursive functions, linear recursion, binary recursion Binary Search Trees: insert, delete, search operations, time complexity analysis of these operations
Suggested Readings (Text/Reference Books):	
1.	Goodrich, M.T., Tamassia, R., & Mount, D., Data Structures and Algorithms Analysis in C++, 2nd edition, Wiley, 2011
2.	Cormen, T.H., Leiserson, C.E., Rivest, R. L., Stein C. Introduction to Algorithms, 4thedition, Prentice Hall of India, 2022
3.	Sahni, S. Data Structures, Algorithms and applications in C++, 2nd edition, Universities Press, 2011

Course Code:	P4	Course Title:	PRACTICAL BASED ON P1 & P3

Course Objectives:

- ✓ Aims to provide students with both theoretical understanding and practical skills in data structure concepts and their implementation using the C++ programming language
- ✓ Develop problem-solving skills through hands-on coding exercises and programming assignments
- ✓ Learn how to identify and fix common programming errors and logical bugs
- ✓ Given the problem statement, students are required to formulate problem, develop Flowchart/algorithm, write code, execute, and test it. Students should be given assignments on following

a) To learn elementary techniques involving arithmetic operators and mathematical expressions, appropriate use of selection (if, switch, conditional operators) and control structures

b) Learn how to use functions and parameter passing in functions, writing recursive programs.

Programming problems based on PROGRAMMING METHODOLOGY

1. Write Programs to learn the use of strings and string handling operations.
   a. Problems which can effectively demonstrate use of Arrays, Structures and Union.
   b. Write programs using pointers.
   c. Write programs to use files for data input and output.
   d. Write programs to implement search algorithms.
   e. Write a program to remove the duplicates from an array.
   f. Write a program that prints a table indicating the number of occurrences of each alphabet in the text entered as command line arguments.
2. 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
3. Write a program to merge two ordered arrays to get a single ordered array.
4. Write a program to search a given element in a set of N numbers using Binary search
(1) with recursion (ii) without recursion.
5. Write a program to calculate GCD of two numbers (i) with recursion (ii) without recursion.
6. 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
7. 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.
8. 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.
9. 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.

Programming problems based on DATA STRUCTURES USING C++

1.Write a program to implement singly linked list as an ADT that supports the following operations:

(i) Insert an element x at the beginning of the singly linked list
(ii) Insert an element x at ith position in the singly linked list
(iii) Remove an element from the beginning of the singly linked list
(iv) Remove an element from ith position in the singly link
(v) Search for an element x in the singly linked list and return its pointer
(vi) Concatenate two singly linked lists

2. Write a program to implement doubly linked list as an ADT that supports the following operations:
(i) Insert an element x at the beginning of the doubly linked list
(ii) Insert an element x at ith position in the doubly linked list
(iii) Insert an element x at the end of the doubly linked list
(iv) Remove an element from the beginning of the doubly linked list
(v) Remove an element from ith position in the doubly linked list.
(vi) Remove an element from the end of the doubly linked list
(vii) 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:
(i) Insert an element x at the front of the circularly linked list
(ii) Insert an element x after an element y in the circularly linked list
(iii) Insert an element x at the back of the circularly linked list
(iv) Remove an element from the back of the circularly linked list
(v) Remove an element from the front of the circularly linked list
(vi) Remove the element x from the circularly linked list
(vii) Search for an element x in the circularly linked list and return its pointer
(viii) 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 which supports the following operations:
(i) Insert an element x
(ii) Delete an element x
(iii) 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
(iv) Display the elements of the BST in preorder, inorder, and postorder traversal
(v) Display the elements of the BST in level-by-level traversal
(vi) Display the height of the BST
10. Write a program to implement a balanced search tree as an ADT.

Course Code:	Q-2	Course Title:	Programming with C++
Course Objectives:			
✓ The course alms at developing basic skills of programming with C++			

UNIT	TOPICS
Unit-I	Introduction to C++ Programming- Character Set, Keywords, Variables: declaration and Initialization, constant and its types, data type, types of operators, operators precedence and associativity, type conversion and typecasting, Basic Program Structure in C++, Input and Output statements
Unit-II	Selection Statements- If, if else, nested if else, else if ladder, switch case and Repetition Statements- for, while, do-while, nesting of loops, break and continue statements
Unit-III	Functions, types of functions, passing values to functions, Recursion and recursive functions, Arrays notation and representation, single and multidimensional arrays
Unit-IV	File Handling: File Classes, opening manipulation and closing a file, file modes manipulation of files, Functions for input output operations

Suggested Readings (Text/Reference Books):

1. Sahni, S. Data Structures, Algorithms and applications in C++, 2nd edition,
2. E. Balagurusamy, Object Oriented Programming with C++, 8th edition, McGraw Hill, 2020
3. Ashok N Kamthane, Object Oriented Programming with ANSI and Turbo C++, Pearson Education 2003

B.Sc.-III Semester (NEP-Revised Syllabus 2024)

Course Code:	P5	Course Title:	WEB DEVELOPMENT ESSENTIALS: HTML, CSS, PHP

Course Objectives:

- ✓ Understanding the basics of Web Designing and usage of various tags along with appropriate styling
- ✓ Designing websites based on different themes, layouts, and functionalities
- ✓ Learn how to use Server-side programming for creating dynamic web services along with customised requirements

UNIT	TOPICS
UNIT- I	Introduction to World wide web: History of WWW, Browser architecture, websites and web portals, static and dynamic websites, servers and its types, frameworks, Introduction to Servers like WAMP, XAMP, Fundamentals of web designing. User Interface and Experience, standard layouts, navigation, sitemaps
Unit-II	Introduction to HTML: Basics of HTML, structure of HTML page, HTML tags, meta tags, HTML forms, buttons, user inputs, frames, tables. New Features in HTML5, audio and video interface, Headers and Footers, articles, HTML semantics
Unit-III	Cascading Style Sheet (CSS): Introduction to CSS, Structure and syntax of CSS, CSS properties: background, color, fonts, borders, positioning, animation, images, wild card etc. New Features in CSS 3.0, CSS3 properties: opacity, reflections, transitions, transformations, animations, Rounded Corners, Web Fonts, Responsive Web design and Media Query
Unit-IV	Introduction to PHP, structure of PHP page, PHP syntax: variables, loops, functions, conditions, array, string and form processing, regular expressions, date and time, file uploads etc. Exception Handling, Session and state management, Cookies, Session Variables.

Suggested Readings (Text/Reference Books):

1. Moseley and Savaliya, Web Technology, Wiley, 2016
2. Robin Nixon, Learning PHP, MySQL, JavaScript, CSS & HTML5, 4th Edition, O' Reilly, 2015
3. HTML 5 Black Book, 2nd Edition, Kogent Learning Solutions Inc, Dreamtech, 2016
4. Steven Holzner, PHP: The Complete Reference, McGraw Hill, 2017
5. Kevin Tatroe & Peter Macnintyre, Programming PHP, 4th Edition, O' Reilly, 2020

Course Code:	P6	Course Title:	OPERATING SYSTEM

Course Objectives:

- ✓ Introduce students to the fundamental concepts of operating systems, including process management, memory management, file systems, and input/output operations.
- ✓ Provide a comprehensive overview of the role of operating systems in managing computer resources and facilitating user interactions.

UNIT	TOPICS
UNIT- I	Introduction to Operating System: History and Evolution of OS, Basic OS functions, Resource Abstraction, Types of Operating Systems- Multiprogramming Systems, Batch Systems, Time Sharing Systems; Operating Systems for Personal Computers, Workstations and Hand-held Devices, Process Control & Real time Systems. Operating System Organization: Processor and User Modes, Kernels, System Calls and System Programs
Unit-II	Process Characterization, Process Scheduling, Non-Pre-emptive and Pre-emptive Scheduling Algorithms. Inter Process Communication and Synchronization, Concurrent and Dependent Processes, Critical Section, Semaphores, Methods for Inter-process Communication, Process Synchronization
Unit-III	Process Management Deadlock Characterization, Necessary and Sufficient Conditions for Deadlock, Deadlock Handling Approaches: Deadlock Prevention, Deadlock Avoidance and Deadlock Detection and Recovery, Banker's algorithm
Unit-IV	Memory Management: Physical and Virtual Address Space; Memory Allocation Strategies- Fixed and -Variable Partitions, Paging, Segmentation, Virtual Memory. File and I/O Management: Directory Structure, File Operations, File Allocation Methods, Device Management

Suggested Readings (Text/Reference Books):

1. Silberschatz, A., Galvin, P. B., Gagne G. Operating System Concepts, 9th edition, John Wiley Publications, 2016
2. Tanenbaum, A. S. Modern Operating Systems, 3rd edition, Pearson Education, 2007
3. Stallings, W. Operating Systems: Internals and Design Principles, 9th edition, Pearson Education, 2018.

Course Code:	Q-3	Course Title:	Web Technology Essentials.

Course Objectives:

- ✓ Understanding the basics of Web Designing and usage appropriate styling various tags along with appropriate styling
- ✓ Designing websites based on different themes, layouts and functionalities
- ✓ Learn how to use Server- side programming for creating dynamic web services along with customised requirements

UNIT	TOPICS
Unit-I	Introduction to World wide web: History of WWW, Browser architecture, Client and server concept, websites and web portals, static and dynamic websites, types of servers and frameworks, Introduction to Servers Fundamentals of web designing, User Interface and Experience, standard layouts, navigation, sitemaps
Unit-II	Introduction to HTML: Basics of HTML, structure of HTML page, HTML tags meta tags, HTML forms, frames, tables, buttons, user inputs. New Features in HTML5, audio and video interface, Headers and Footers, articles, HTML semantics
Unit-III	Cascading Style Sheet (CSS): Introduction to CSS, Structure and syntax of CSS, properties: background, colour, fonts, borders, positioning, animation, images, wild card etc.
Unit-IV	New Features in CSS 3.0 CSS3 properties: opacity, reflections, transitions transformations animations, Rounded Corners, Web Fonts, Responsive Web design and Media Query

Suggested Readings (Text/Reference Books):

1.	Moseley and Savaliya, Web Technology, Wiley, 2016
2.	Robin Nixon, Learning PHP, MySQL, JavaScript, CSS & HTML5, 4th Edition, O' Reilly, 2015
3.	HTML 5 Black Book, 2nd Edition, Kogent Learning Solutions Inc, Dreamtech, 2016

UNIVERSITY OF LUCKNOW
DEPARTMENT OF COMPUTER SCIENCE
Lucknow, Uttar Pradesh PIN-226007

B.Sc.-V Semester (NEP-Revised Syllabus 2024)

Course Code:	P9	Course Title:	COMPUTER ARCHITECTURE & MICROPROCESSOR

Course Objectives:

- ✓ This course Introduces students to the fundamental concepts of digital computer organization, design, and architecture. It aims to develop a basic understanding of the building blocks of a computer system and highlights how these blocks are organized together to architect a digital computer system.
- ✓ This course introduces the internal architecture, programming models of intel Microprocessors (8086-Pentium) and assembly language programming.

UNIT-	TOPICS
UNIT- I	Digital Logic Circuits: Digital Logic Gates, Flip flops and their characteristic table, Logic circuit simplification using Boolean algebra and Karnaugh map. Don't care conditions, Combinational circuits
Unit-II	Sequential Circuits: Digital Components: Decoders, Encoders, Multiplexers, De-Multiplexers, Adder (Half and full adder), Subtractor, Binary Incrementor, Registers, and Memory Units
Unit-III	Microprocessor Architecture (8086): Internal Architecture, BIU and EU Programming Model, Addressing Modes, Instructions Set, Types of Bus Bus timings, Memory address decoding, cache memory and cache controllers, I/O interface, timer, Interrupt controller, DMA controller, video controllers, communication interfaces.
Unit-IV	Microprocessor programming: Register Organization, instruction formats, Program control instructions through assembly language

Suggested Readings (Text/Reference Books):

1. Computer Architecture: A Quantitative Approach, John L. Hennessy, David A. Patterson, 4th Edition.
2. Computer Organization and Architecture, William Stallings, Prentice Hall.
3. Brey, B.B. The Intel Microprocessors: Architecture, Programming and Interfacing, 8th edition, Pearson education, 2009.
4. Triebel, W.A., & Singh, A. The 8088 and 8086 Microprocessors Programming, Interfacing, Software, Hardware and Applications, 4th edition, Pearson education, 2002.

Course Code:	P10	Course Title:	PROGRAMMING IMPLEMENTATION WITH PYTHON PROGRAMMING

Course Objectives:

- ✓ The course introduces students to the concept of data handling using files and GUI designing. This would equip the students with knowledge to work on real world data from various applications and GUI development for effective data handling.

UNIT-	TOPICS
UNIT- I	Introduction to Python, Python, Features of Python, Execution of a Python, Program, Writing Our First Python Program, Data types in Python. Python Interpreter and Interactive Mode; Values and Types, Variables, Expressions, Statements, Tuple Assignment, Precedence of Operators, Comments; Modules and Functions, Function Definition and use, Flow of Execution, Parameters and Arguments
Unit-II	Operators in Python, Input and Output, Control Statements. Boolean Values and operators, Conditional (if), Alternative (if-else), Chained Conditional (if- elif-else); Iteration: while, for, break, continue, pass; Fruitful Functions: Return Values, Parameters, Local and Global Scope, Function Composition, Recursion
Unit-III	Arrays in Python, Strings and Characters. Strings: String Slices, Immutability, String Functions and Methods, String Module; Lists as Arrays. Exponentiation, Linear Search, Binary Search. Functions, Lists and Tuples. List Operations, List Slices, List Methods, List Loop, Mutability, Aliasing, Cloning Lists, List Parameters;
Unit-IV	Tuples: Tuple Assignment, Tuple as Return Value; Dictionaries: Operations and Methods; Advanced List Processing - List Comprehension Files and Exception: Text Files, Reading and Writing Files, Format Operator; Command Line Arguments, Errors and Exceptions, Handling Exceptions, Modules, Packages;

Suggested Readings (Text/Reference Books):

1.	Mark Lutz, Learning Python
2.	Tony Gaddis, Starting Out with Python
3.	Kenneth A. Lambert, Fundamentals of Python

The students are required to verify their ability to use core programming basics and program design with functions using Python programming language. The teacher shall program to strengthen the practical expertise of the students. The following is an indicative list of programs that can be practiced

1.Write a program to demonstrate different number data types in Python.
2.Write a program to perform different Arithmetic Operations on numbers in Python,
3. Write a program to create, concatenate and print a string and accessing sub-string from a given string.
4. Write a python script to print the current date in the following format "Fri Oct 11 02:26:23 IST 2019"
5. Write a program to create, append, and remove lists in python.
6. Write a program to demonstrate working with tuples in python.
7.Write a program to demonstrate working with dictionaries in python.
8.Write a python program to find largest of three numbers.
9. Write a Python program to construct the following pattern, using a nested for loop

```
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

10. Write a Python script that prints prime numbers less than 20.
11.Write a python program to define a module to find Fibonacci Numbers and import the module to another program.
12.Write a python program to define a module and import a specific function in that module to another program.
13. Write a program that inputs a text file. The program should print all the unique words in the file in alphabetical order.
14. Write a Python class to convert an integer to a roman numeral.
15.Write a Python class to reverse a string word by word.
1.

B.Sc.-VI Semester (NEP-Revised Syllabus 2024)

Course Code:	P11	Course Title:	COMPUTER GRAPHICS

Course Objectives:

- ✓ This course introduces fundamental concepts of Computer Graphics with a focus on modeling, rendering and interaction aspects of computer graphics. The course emphasizes the basic principles needed to design, use and understand computer graphics systems.

UNIT5	TOPICS
UNIT- I	Introduction: Introduction to Graphics systems, Basic elements of Computer graphics, Applications of computer graphics. Architecture of Raster and Random scan display devices, Input/output devices.
Unit-II	2-D Geometrical Transforms: Translation, Scaling, Rotation, Reflection and Shear Transformations, Composite Transforms, Transformations between Coordinate Systems. 2-D Viewing: The Viewing Pipeline, Viewing Coordinate Reference Frame, Window to Viewport Coordinate Transformation, Viewing Functions
Unit-III	Line Clipping Algorithms- Cohen-Sutherland and Cyrus Beck Line Clipping Algorithms, Sutherland-Hodgeman Polygon Clipping Algorithm. 3-D Object Representation: Polygon Surfaces, Quadric Surfaces, Spline Representation 3-D Geometric Transformations: Translation, Rotation, Scaling, Reflection and Shear Transformations, Composite Transformations, 3-D Viewing: Viewing Pipeline, Viewing Coordinates, View Volume, General Projection Transforms and Clipping.
Unit-IV	Virtual Reality: Basic Concepts, Classical Components of VR System, Types of VR Systems, Three-Dimensional Position Trackers, Navigation and Manipulation Interfaces, Gesture Interfaces. Input Devices, Graphical Rendering Pipeline, Haptic Rendering Pipeline, Open GL Rendering Pipeline. Applications of Virtual Reality.

Suggested Readings (Text/Reference Books):

1.	Donald Hearn and M. Pauline Baker, "Computer Graphics with Open GL", Prentice Hall,
2.	R. K Maurya, "Computer Graphics with Virtual Reality", Wiley
3.	"Computer Graphics Principles & practice", Foley, Van Dam, Feiner and Hughes, Pearson Education.

Course Code:	P12	Course Title:	COMPUTER NETWORK

Course Objectives:

- ✓ 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.

UNIT	TOPICS
Unit-I	Introduction to Computer Networks and Networking Elements: Network Definition, Network Topologies, Network Classifications, Network Protocol, Layered Network Architecture, Overview of OSI Reference Model, Layers and services, Overview of TCP/IP Protocol Suite, Hub, Switch (Managed and Unmanaged), Routers
Unit-II	Data Communication Fundamentals and Techniques: Analog and Digital Signal, Data-Rate Limits, Digital to Digital Line Encoding Schemes, Pulse Code Modulation, Parallel and Serial Transmission, Digital to Analog Modulation Multiplexing Techniques- FDM, TDM, Transmission Media.
Unit-III	Networks Switching Techniques and Access Mechanisms: Circuit Switching, Packet Switching- Connectionless Datagram Switching, Connection-Oriented Virtual Circuit Switching; Dial-Up Modems, Digital Subscriber Line, Cable TV for Data Transfer.
Unit-IV	Multiple Access Protocol and Network Layer: CSMA/CD Protocols, Ethernet LANS; Connecting LAN and Back-Bone Networks- Repeaters, Hubs, Switches, Bridges, Router and Gateways, Networks Layer Functions and Protocols Routing, Routing Algorithms, Network Layer Protocol of Internet - IP Protocol, Internet Control Protocols. Transport Layer and Application Layer Functions and Protocols: Transport Services- Error and Flow Control, Connection Establishment and Release- Three Way Handshake, Application Layer Protocol

Suggested Readings (Text/Reference Books):

1. B. A. Forouzan : Data Communications and Networking, fourth edition, THM Publishing Company Ltd 2007.
2. A. S. Tanenbaum: Computer Networks, Fourth edition, PHI Pvt. Ltd 2002

Course Code:	P13-A	Course Title:	ARTIFICIAL INTELLIGENCE

Course Objectives:

- ✓ To introduce basic concepts and techniques of Artificial Intelligence (Al).
- ✓ To apply informed search techniques for different applications.
- ✓ To learn about the latest techniques for developing At systems.

UNIT	TOPICS
Unit-I	Introduction to Artificial Intelligence: Definition of Al; Turing Test; Brief History of Al. Problem Solving and Search: Problem Formulation; Search Space; States vs. Nodes; Tree Search: Breadth-First, Uniform Cost, Depth-First, Depth-Limited, Iterative Deepening; Graph Search.
Unit-II	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-III	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-IV	AI The Present and the Future: Symbolic Al, Data-driven Al and Machine Learning, Introduction to Machine Learning and Deep Learning based Al, some applications of symbolic and data driven Al, Interpretable and Explainable Al, Ethics of Al: benefits and risks of Al.

Suggested Readings (Text/Reference Books):

1.	Elaine Rich, Kevin Knight, Shivashankar B Nair, Artificial Intelligence, Third Edition, McGraw Hill Edition
2.	Russell Stuart Jonathan and Norvig Peter, Artificial Intelligence: A Modern Approach, 3rd Edition, Prentice Hall, 2010
3.	Bratko, Ivan, Prolog Programming for Artificial Intelligence, Addison-Wesley, Pearson Education, 4th edition, 2012.

Course Code:	P13-B	Course Title:	DESIGN ANALYSIS AND ALGORITHM

Course Objectives:

- ✓ Introduce students to various algorithm design paradigms such as divide and conquer, dynamic programming, greedy algorithms, and graph algorithms.
- ✓ Provide insight into the strengths, weaknesses, and applications of different algorithmic approaches.

UNIT	TOPICS
Unit-I	Introduction: Basic Design and Analysis Techniques of Algorithms, Correctness of Algorithm. Algorithm Design Techniques: Iterative Techniques, Divide and Conquer, Dynamic Programming, Greedy Algorithms.
Unit-II	Sorting and Searching Techniques: Elementary Sorting techniques- Bubble Sort, Insertion Sort, Merge Sort, Advanced Sorting techniques Heap Sort, Quick Sort, Sorting in Linear Time Bucket Sort, Radix Sort and Count Sort, Searching Techniques- Medians & Order Statistics, complexity analysis
Unit-III	Graphs Algorithms: Graph Algorithms- Breadth First Search, Depth First Search and its Applications, Minimum Spanning Trees. String Processing
Unit-IV	Lower Bounding Techniques: Decision Trees, Balanced Trees, Red-Black Trees Advanced Analysis Technique: Randomized Algorithm, Distributed Algorithm, Heuristics

Suggested Readings (Text/Reference Books):

1.	T.H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein Introduction to Algorithms, PHI, 3rd Edition 2009
2.	Sara basse& A.V. Gelder Computer Algorithm - Introduction to Design and Analysis, Publisher-Pearson 3rd Edition 1999

Course Code:	P13-C	Course Title:	SOFTWARE ENGINEERING

Course Objectives:

- ✓ This course will acquaint the student with different approaches and techniques used to develop good quality software. The course includes learning of software development process frameworks, requirement analysis, design modelling, qualitative and quantitative software metrics, risk management, and testing techniques.

UNIT	TOPICS
Unit-I	Software Development Approaches: Introduction, Evolving Role of Software; Software Characteristics: Software Applications. Software Design Processes: Introduction, Definitions of Software Engineering: The Serial or Linear Sequential Development Model: Iterative Development Model ;Incremental Development Model
Unit-II	Software Design Principles: Introduction, System Models: Data-flow Models, Semantic Data Models, Object Models, Inheritance Models, Object Aggregation, Service Usage Models, Data Dictionaries; Software Design: The Design Process, Design Methods, Design description, Design Strategies, Design Quality; Architectural Design: System Structuring. The Repository Model, The Client-Server Model, The Abstract Machine Model, Control Models, Modular Decomposition, Domain-Specific Architectures.
Unit-III	Object Oriented Design: Introduction; Object Oriented Design: Objects, Object Classes & Inheritance, Inheritance, Object Identification, An Object-Oriented Design Example, Object Aggregation; Service Usage; Object Interface Design: Design Evolution, Function Oriented Design, Data-Flow Design; Structural Decomposition: Detailed Design
Unit-IV	Software Reliability: Introduction; Software Reliability Metrics; Programming for Reliability: Fault Avoidance, Fault Tolerance, Software Reuse. Software Testing Techniques: Introduction; Software Testing Fundamental; Testing Principles; White Box Testing; Control Structure Testing; Black Box Testing; Boundary Value Analysis; Testing GUIs; Testing Documentation and Help Facilities; Software Testing Strategies: Introduction; Organizing for Software Testing: Software Testing Strategy, Unit Testing: Unit Test Considerations, Top-Down Integration, Bottom-Up Integration.

Suggested Readings (Text/Reference Books):

1.	R. G. Pressman - Software Engineering, TMHSommerville, lan, Software Engineering Pearson Education
2.	Pankaj Jalote -An Integrated Approach Publications. to Software Engineering, Narosa Publications.
3.	Pfleeger, Shari Lawrence, Software Engineering Theory and Practice, second edition. Prentice-Hall 2001.
4.	Object Oriented & Classical Software Engineering (Fifth Edition), SCHACH, TMH