SEMESTER I

PRINCIPLES OF PROGRAMMING WITH C

Core Course - II	24UCP401	Principles of Programming with C	Credits - 4

Course Outcomes: At the end of the course the student will be able to

CO1	Elucidate the basic architecture and functionalities of a Computer
CO2	Understand the programming paradigm of C programming languages
CO3	Implement the concept of arrays, strings and functions
CO4	Apply code reusability with functions and pointers
CO5	Develop C programs using structure, union and file concepts

SYLLABUS

UNIT I: Principles of Computer & Overview of C

Introduction - Components of a Computer – Concept of Hardware and Software – Art of Programming through Algorithms and Flowcharts - History of C – Importance of C – Sample Programs – Basic Structure of C programs– Executing C Program.

UNIT II: Constants, Decision Making and Looping

Introduction – Character set – C Tokens - Constants, Variables and Data Types – Declaration of Variables - Declaration of Storage Class – Symbolic Constants – Operators and Expressions –Reading and Writing a Character – Formatted Input – Formatted Output - Decision Making and Branching – Decision Making and Looping.

UNIT III: Arrays & Functions

Arrays – one dimensional array – Declaration & Initialization and access- 2D arrays and multidimensional arrays- Strings – defining strings, initializing, accessing String handling functions- Functions Introduction – Elements of User – defined Functions – Definition of Functions – Function Declaration – Category of Functions – Nesting of functions - Recursion – Passing Arrays to Functions.

UNIT IV: Structure & Union

Structures: Introduction – Defining a structure – Declaration of structure – Accessing Structures members – Array of Structures – Arrays within Structures - Structures within structures – Structures and functions – Union.

UNIT – V: Pointers & Files

Introduction – Declaring pointer variable – Initialization of Pointer variables - Chain of Pointers - Pointer Expressions —Pointers and Arrays – Array of Pointers – Pointers as function arguments – Functions returning Pointers – Pointers to Functions – Pointers and Structures - Files: Introduction – Defining and Opening a File – Closing a File – Input / Output operations on files.

Text Books:

1. E. Balagurusamy, “Programming in ANSI C”, McGraw Hill, 8th Edition, 1st January 2024.

Reference Books:

1. Jyoti P Mirani & Vikas B. Agarwal, "Computer Fundamentals", 2nd Edition, Nirali Prakashan, 2019.
2. Yashavant Kanetkar, "Let Us C", 18th Edition, BPB Publications, 2021.
3. Szuhay, Jeff, and Szuhay, Jeff, "Learn C Programming: A Beginner's Guide to Learning C Programming the Easy and Disciplined Way", Packt Publishing, 2020.
4. Jena, Sisir Kumar, and Jena, Sisir Kumar, "C Programming: Learn to Code", 1st Edition, CRC Press, 2021.
5. Reema Thareja , "Programming in C", 3rd Edition, Oxford University Press, 2023.

PRINCIPLES OF PROGRAMMING WITH CLAB

Course Outcomes: At the end of the course the student will be able to

CO1	Remember and understand the logic of given concept and construct the programs
CO2	Apply the concept of single and multi-dimensional arrays to solve problems related to searching, sorting and matrix operations
CO3	Apply the concept of Strings for writing programs related to character array.
CO4	Write programs using concept of user defined and recursive functions.
CO5	Make use of different data-structures like arrays, pointers, structures and files

SYLLABUS

1. Write C program
   (i) To convert temperature from degree Centigrade to Fahrenheit,
   (ii) Find whether given number is even or odd,
   (iii) Find the greatest of three numbers.
2. Write a C program to find the sum, average, standard deviation for a given set of numbers.
3. Write a C program to print magic square of order n where n>3 and n is odd.
4. Write C program to solve quadratic equation using functions.
5. Write C program
   (i) To swap two numbers using call by value
   (ii) To change the elements of an array using call by reference
6. Write C program to count the number of vowels in the given sentence.
7. Write C program to find the factorial of a given number using a recursive function.
8. Write C program to reverse a number using pointer.
9. Write C program to check whether the given string is a palindrome using pointers.
10. Write C program to print the students mark sheet assuming roll no, name, and marks in 5 subjects in a structure. Create an array of structures and print the mark sheet in the university pattern.
11. Write a function using pointers to add two matrices and to return the resultant matrix to the calling Function.
12. Write a C program which receives two file names as arguments and check whether the file contents are same or not. If same delete the second file.

DIGITAL COMPUTER FUNDAMENTALS

Core Course - I	**24UCP402**	**Digital Computer Fundamentals**	**Credits - 4**

COURSE OUTCOMES: At the end of the course the student will be able to

CO1	Understand various types of number systems and their conversions
CO2	Familiar with the Boolean algebra, logic gates, logical variables and its the truth table
CO3	Design and implement variety of logical devices using combinational circuits concepts.
CO4	Analyze sequential circuits like Registers and Counters using flip-flops
CO5	Demonstrate and compare the construction of programmable logic devices and different types of ROM

SYLLABUS

UNIT I: Number Systems

Digital Systems– Binary numbers – Number base conversions- Octal, Hexadecimal -one's complement and two's complement, Binary Arithmetic, Fixed point representation and Floating point representation of– Signed Binary Numbers - Binary Codes

UNIT II: Boolean Algebra & Logic Gates

Boolean Algebra –Basic definitions – Axiomatic Definition of Boolean Algebra -Basic Theorems and Properties of Boolean Algebra – Boolean Functions- Canonical and standard forms –The AND gate, The OR gate, The NAND gate, NOR gate, The Exclusive–OR gate and Exclusive-NOR gate- Boolean Algebra and Logic Simplification- DeMorgan's Theorems, Boolean Expressions and Truth Tables, The Karnaugh Map, SOP, POS, Don't care conditions.

UNIT – III: Combinational Logic Circuits

Basic combinational Logic Circuits, Implementing Combinational Logic, The Universal Property of NAND and NOR Gates. Functions of Combinational Logic - Adders, Subtractors, Decoders, Encoders, Code Converters, Multiplexers, Demultiplexers.

UNIT – IV: Sequential Circuit Design

Flip flops (RS, Clocked RS, D, JK, JK Master Slave, T, Counters, Shift registers and their types, Counters: Synchronous and Asynchronous counters.

UNIT – V: Memory and Programmable Logic

Introduction, Random access memory, memory decoding, error detection and correction, read only memory, Flash Memory - Special Types of Memories, Magnetic and Optical Storage- programmable logic array, sequential programmable devices.

Text Book :

1. M. Morris Mano, Michael D. Ciletti, "Digital Design", 6th Edition, Pearson Education, 2018.

Reference Book :

1. Donald P leach, Albert Paul Malvino, Goutam Saha, Digital Principles and Application, 8th Edition., McGraw Hill education (India) Private Limited, 2015.
2. Prof. Satish Jain Shashi Singh, "Digital Computer Fundamentals", 1st Edition, BPB Publications, 2017.
3. Ronald Tocci, Neal Widmer , Gregory Moss," Digital Electronics Principles and applications,12th Edition, Pearson, 2017.
4. Digital Electronics Circuits and Systems, Bishnu Charan Sarkar and Suvra Sarkar,1st Edition, Sad-Ul-Tan Prakasani, Burdwan, 2018.
5. Roger L. Tokheim, Patrick E., "Digital Electronics: Principles and Applications", 9th Edition, McGraw-Hill Education, 2022.

SEMESTER II

CORE COURSE - III
OBJECT ORIENTED PROGRAMMING WITH JAVA
COURSE CODE: 24UCA504

SYLLABUS

Unit I

The History and the Evolution of Java - The Byte code - The Java Buzzwords- An overview of Java: Object Oriented Programming- Simple Program- Two Control Statements- Using Blocks of Code- Lexical Issues- The Java class libraries- Data Types, Variables and Arrays: Java is a strongly typed language- The Primitive types- Integers- Floating Point Types- Characters- Booleans- Literals- Variables- Type Conversion and Casting- Arrays-Operators.

Unit II

Control Statements: Java's selection statements- Iteration statements- jump statements- Introducing Classes: Class fundamentals- Declaring Objects- Assigning Object Reference Variables- Introducing Methods- Constructors- The this keyword- Garbage Collection- The finalize() method- Overloading Methods- Using Objects as Parameters- Returning Objects- Recursion- Introducing Access Control- Understanding Statics- Introducing Final.

Unit III

Inheritance: Inheritance Basics- Using Super- Creating Multi Level Hierarchy- When Constructors are Called- Method Overriding- Dynamic Method dispatch- Using Abstract Classes- Using Final with Inheritance-Packages and Interfaces: Packages- Access Protection- Importing Packages-Interfaces.

Unit IV

Exception Handling: Exception Handling Fundamentals- Exception types- Uncaught Exceptions- Using Try and Catch- Multiple Catch Clauses- Nested Try Statements- Throw- Throws- Finally- Java's Built-in Exception- Creating your own exception subclasses- Multithreaded Programming: The Java Thread Model- The Main Thread- Creating a Thread- Creating Multiple Threads- Using isAlive() and join()-Thread Priorities- Synchronization- Interthread Communication.

.

Unit V

Input/Output: Exploring Java.io - The Java I/O classes and Interfaces- File- The Stream Classes- The Byte Streams- the Character Streams- The Applet Class- Two Types of Applets- Applet Basics- Applet Architecture- An Applet Skeleton- Simple Applet Display Method- Requesting Repainting- Using the Status Window- The HTML APPLET Tag-Passing Parameter to Applets - getDocumentBase() and getCodeBase()- AppletContext and showDocument().

CORE LABORATORY - II
OBJECT ORIENTED PROGRAMMING WITH JAVA LAB
COURSE CODE: 24UCA551

SYLLABUS

1. Write a program to sort the given numbers using arrays.
2. Write a program to implement the FIND and REPLACE operations in the given multiple text.
3. Write a program to implement a calculator to perform basic arithmetic Operations.
4. Write a program to find the area of a rectangle using constructor.
5. Write a program to find the student's percentage and grade using command line arguments.
6. Write a program for factorial using recursion.
7. Write a program to draw circle or triangle or square using polymorphism and inheritance.
8. Implement multiple inheritance concepts in java using interface, you can choose your own example of a company or education institution or a general concept which requires the use of interface to solve a particular problems.
9. Write a java program using packages.
10. Write a program to create threads and assign priorities to them.
11. Write a program to develop an applet to play multiple audio clips using multithreading.
12. Write a program to create a window with three check boxes called red, green and blue. The applet should change the colors according to the selection.
13. Write a program to retrieve employee data from a file
14. Write a program to retrieve student data from a Database.
15. Write a program to develop applications involving File handling.

SEMESTER III

Core Course – III
24UCP403: Python Programming

Credits: 4 **Internal Marks: 25**
Contact Hours: 4 **External Marks: 75**

Course Outcomes:

Unit No.	CO. No.	Upon completion of this course, the students will be able to	Cognitive Level
Unit - I	CO1	Understand the basic building blocks of programming constructs	K1, K2
Unit - II	CO2	Apply control structures such as loops, conditionals, and iterators to solve basic computational problems.	K1 – K4
Unit - III	CO3	Develop modular Python programs using functions, handle files and exceptions effectively, and use standard libraries.	K1 – K4
Unit - IV	CO4	Implement object-oriented programming concepts such as classes, objects, methods, encapsulation, polymorphism, and operator overloading.	K1 – K4
Unit - V	CO5	Analyze and work with scientific computing libraries like NumPy and pandas to perform efficient data manipulation and numerical operations.	K1 – K4

Syllabus

Unit I

Introducing Python - Python on Different Operating Systems: Python on Windows- Python on macOS - Python on Linux - Variables and Assignment Statements – Arithmetic – Function print and an Introduction to Strings.

Unit II

Decision Making: The if Statement and Comparison Operators - Objects and Dynamic Typing-Control Statements: If Statement-If Else and If Elif Else Statements -While Statement, For Statement -Iterables, Lists and Iterators -Builtin Range Function -Augmented Assignments –Sequences: Lists and Tuples-Dictionaries and Set-controlled Iteration; Formatted Strings –Sentinel controlled Iteration - Builtin Function Range-Break and Continue Statements.

Unit III

Functions: Defining Functions-Functions with Multiple Parameters- Python Standard Library -math Module Functions - Using IPython Tab Completion for Discovery - Default Parameter Values -Keyword Arguments -Arbitrary Argument Lists- Methods: Functions That Belong to Objects- Scope rules- Recursion- Files: Text File Processing - Updating Text Files - Handling Exceptions- finally Clause -Explicitly Raising an Exception.

Unit IV

Object Orientation: Custom Class Account- Definition-Composition: Object References as Members of Classes- Controlling Access to Attributes-Properties for Data Access- Account Class Definition-Composition: Object References as Members of Classes -Controlling Access to Attributes-Properties for Data Access- Duck Typing and Polymorphism- Operator Overloading- Namespaces and Scopes.

Unit V

NumPy Basics: Arrays and Vectorized Computation: The NumPy ndarray: A Multidimensional Array Object- Pseudorandom Number Generation- Universal Functions: Fast Element-Wise Array Functions- Array-Oriented Programming with Arrays- File Input and Output with Arrays- Linear Algebra- Getting Started with pandas: Introduction to pandas Data Structures- Essential Functionality.

Text Book:

1. Paul Deitel and Harvey Deitel, “Python for Programmers”, Pearson Education, 1st Edition, 2021(Unit-I-IV)
2. Python for Data Analysis Data Wrangling with pandas, NumPy, and Jupyter, Wes McKinney, oreilly ,3rd Edition,2022(Unit-V)

Reference Books:

1. Python Crash Course by Eric Matthes, 3rd Edition, No Starch Press, January 2023
2. Martin C. Brown, Python: The Complete Reference, McGraw Hill Education (India), 2024.
3. Jeeva Jose and P. SojanLal, “Introduction to Computing and Problem Solving with PYTHON”, Khanna Book Publishing Co, 2016.
4. Phill Vega., "Python Programming 2020: Guide to Learn Python, Step by Step, with Practical Exercises, Tips and Tricks”, Independently Published, 2020.
5. Reema Thareja,"Python Programming: Using Problem Solving Approach”, Oxford University Press, 2019.
6. Kanetkar Aditya Yashavant, Kanetkar, Let Us Python, 2nd Edition, BPB Publications, 2020.

Core Laboratory – III
24UCP452: Python Programming Lab

Credits: 2 **Internal Marks: 40**
Contact Hours: 4 **External Marks: 60**

Course Outcomes:

CO. No.	Upon completion of this course, the students will be able to	Cognitive Level
CO1	Understand and explain Python syntax, data types, operators, and control structures.	K3
CO2	Apply functions, loops, and conditional statements to develop modular and logical Python programs.	K3
CO3	Analyze problems and implement appropriate data structures such as lists, dictionaries, and classes.	K4
CO4	Develop Python applications using object-oriented concepts and file handling techniques.	K4
CO5	Implement algorithms for sorting, searching, and graph traversal and demonstrate their functionality.	K3,K4

Syllabus

1. Write a Python program to simulate an ATM using conditional statements.
2. Write a Python program for a BMI calculator with health condition output.
3. Write a Python program to count word frequency in a sentence.
4. Write a Python program to compute the factorial of a number using both loops and recursion.
5. Write a Python program for an inventory management system using dictionary.
6. Write a Python program for an employee attendance tracker using dictionary.
7. Write a Python program to add two matrices of the same size and display the result.
8. Write a Python program to find the transpose of a matrix.
9. Write a Python program for a bank transaction system using functions.
10. Write a Python program for a to-do list application using file handling.
11. Create a program to count the number of times a specific word appears in a given text file.
12. Implement a BankAccount class with methods to deposit, withdraw, and display the balance.
13. Create a Vehicle class and derive Car and Bike subclasses from it.
14. Write a Python program Basic Array Operations using NumPy
15. Write a Python program to create and display a DataFrame using Pandas.
16. Write a Python program to implement a Stack and queue data structure using a list. Include push, pop, peek, and is_empty operations.
17. Write a Python program to implement a singly linked list with the following operations:
    Insertion at the end
    Deletion of a node by value
    Display the list
18. Write a Python program to implement the following sorting algorithm using Bubble Sort.
19. Write a Python program to implement Linear Search and Binary Search.
20. Write a Python program to implement graph traversal using Breadth-First Search (BFS) and Depth-First Search (DFS).

Core Course - V
24UCP404: Data Structures and Algorithms

Credits: 4 **Internal Marks: 25**
Contact Hours: 4 **External Marks: 75**

Course Outcomes:

Unit No.	CO. No.	Upon completion of this course, the students will be able to	Cognitive Level
Unit - I	CO1	Explain the concept of algorithms and analyze their performance using asymptotic notations.	K1 – K4
Unit - II	CO2	Implement and use linear data structures (arrays, stacks, queues, linked lists).	K1 – K4
Unit - III	CO3	Apply tree and graph-based data structures for hierarchical and networked data modelling.	K1 – K4
Unit - IV	CO4	Design and implement efficient sorting and searching algorithms using Divide and Conquer.	K1 – K4
Unit - V	CO5	Solve optimization problems using Greedy and Backtracking techniques.	K1 – K4

Syllabus

Unit I

Introduction of Algorithms, Analysing Algorithms: Theoretical Foundation, Algorithms and it's Specification, Random Access Machine Model ,Counting Primitive Operations, Notion of best case, average case and worst case, Characterizing Run Time, Use of asymptotic notation, Big-Oh Notation, Little-Oh, Omega and Theta Notations.

UNIT II

Arrays: Sparse Matrices–Representation of Arrays- Stacks and Queues - Fundamentals–Evaluation of Expression Infix to Postfix Conversion - Multiple Stacks and Queues - Linked List: Singly Linked List - Linked Stacks and Queues – Polynomial Addition- More on Linked Lists – Sparse Matrices – Doubly Linked List and Dynamic – Storage Management –Garbage Collection and Compaction.

UNIT III

Basic Terminology – Binary Trees – Binary Tree Representations – Binary Trees-Traversal-More on Binary Trees – Threaded Binary Trees – Binary Tree.Representation of Trees – Counting Binary Trees. Graphs: Terminology and Representations-Traversals, Connected Components and Spanning Trees, Shortest Paths and Transitive Closure.

UNIT – IV

Introduction to Algorithms - Priority Queues - Heaps – Heap Sort – Divide and Conquer Algorithms - Binary Search - Merge Sort – Quick Sort — Finding the Maximum and Minimum.

UNIT – V

Greedy Method: The General Method – Knapsack Problem – Minimum Cost Spanning Trees– Optimal Merge Patterns- Single-Source Shortest Paths. Back Tracking: The General Method – The 8-Queens Problem – Sum of Subsets – Graph Coloring – Hamiltonian Cycles.

Text Book(s):

1. Ellis Horowiz, SartajSahni, "Fundamentals of Data Structure",Galgotia Publications(UNIT I-III)
2. Ellis Horowiz, SartajSahni and SanguthevarRajasekaran, "Computer Algorithms", University Press. (UNIT IV&V)

Reference Book(s):

3. Seymour Lipschutz, "Data Structures",TataMcgraw Hill, Schaum's Outline Series, 2014.

4. Micheal T. Goodrich and Roberto Tamassia: Algorithm Design: Foundations, Analysis and Internet examples (John Wiley & Sons, Inc., 2006).
5. ThomasH.Cormen,ChalesE.Leiserson,RonaldL.Rivest,CliffordStein,–Introduction to Algorithms‖,3rdEdition,McGrawHill, 2010.

SEMESTER IV

WEB DESIGN - 24DCA104

Syllabus

Unit I : Introduction to the Internet
Computers in Business – E Mail - World Wide Web – Internet Technologies – Internet Addressing – Physical Connections – Internet Browser and its types

Unit II : Introduction to HTML
Designing a Home page – History of HTML – HTML Generations – HTML Documents – Anchor Tag – Hyper Links – Head & Body sections – Colourful web pages – Comment Lines

Unit III: Formatting Tags
Heading printing – Aligning the Headings – Horizontal Rule – Paragraph – Tab Settings – Images and Pictures – Ordered List – Unordered List – Nested Lists

Unit IV: Table Handling
Tables – Table Creation in HTML – Width of the Table and Cells – Cells Spanning Multiple Rows and Columns – Coloring Cells – Column Specification

Unit V: DHTML and Style Sheets
Defining Styles – Elements of Styles – Linking a Style Sheet to an HTML Document – In line Styles – External Style Sheets – Internal Style Sheets – Multiple Styles – Web page design & publishing

WEB DESIGN LAB - 24DCA154

Syllabus

1. Design a page having suitable background colour and text colour with title “My First Web Page” using all the attributes of the Font tag.
2. Create a HTML document giving details of your [Name, Age], [Address, Phone] and [Register Number, Class] aligned in proper order using alignment attributes of Paragraph tag.
3. Write HTML code to design a page containing some text in a paragraph by giving suitable heading
4. Create a page to show different character formatting (B, I, U, SUB, SUP) tags.
5. Write HTML code to create a Web Page that contains an Image at its centre.
6. Create web Pages using Anchor tag with its attributes for external links
7. Write a HTML code to create a web page with pink color background and display moving message in red color.
8. Create a web page, showing an ordered list of all second semester courses (Subjects) Write a HTML document to demonstrate Formatting tags.
9. Develop HTML document to demonstrate Ordered lists, unordered Lists, definition Lists.
10. Write an HTML document to create table header rows, data rows, caption and Attributes of the table tag.
11. Create an HTML document to cell padding and cell spacing, Bgcolor, Colspan and Row span attribute.
12. Develop an HTML document using frameset and the targeting named frames.

SEMESTER V

Discipline Centric Core Course - III
24UCA503: SQL and Contemporary Database Techniques

Credits: 4 **Internal Marks: 25**
Contact Hours: 4 **External Marks: 75**

Course Outcomes:

Unit No.	CO. No.	Upon completion of this course, the students will be able to	Cognitive Level
Unit I	CO1	Explain database fundamentals and relational model concepts.	K1 – K4
Unit II	CO2	Design relational databases using ER model and normalization.	K2 – K4
Unit III	CO3	Apply SQL queries for data definition and data manipulation.	K1 – K4
Unit IV	CO4	Describe NoSQL architecture, operations, and consistency models.	K1 – K4
Unit V	CO5	Use MongoDB for data modeling, CRUD operations, and querying.	K1 – K3

Syllabus

UNIT I – Introduction to Database System and Relational Model
Introduction: Database System Applications – Purpose of Database System – View of Database – Database Languages – Database design – Database Engine – Database and Application Architecture – Database Users and Administrators – Introduction to Relational Model: Structure of Relational Databases – Database Schema – Keys – Schema Diagram – Relational Query Languages – Relational Algebra.

UNIT II – ER Model and Relational Database Design
Database Design using the ER Model: Overview of design process – The Entity Relationship Model – Complex Attributes – Mapping Cardinalities – Primary Key – Removing redundant Attributes in Entity Sets – Reducing ER Diagrams to Relational Schemas - Relational Database Design: Decomposition using Functional Dependencies – Normal Forms – Functional Dependency Theory.

UNIT III – SQL and Intermediate SQL
Introduction to SQL: Overview of the SQL Query Language – SQL Data Definition – Basic Structure of SQL Queries – Additional basic operations – Set operations – Null Values – Aggregate Functions – Nested Sub queries – Modification of the Database. Intermediate SQL: Join expressions – Views – Transactions – Integrity Constraints.

UNIT IV – NoSQL Databases: Architecture, Operations, and Consistency
Introduction to NoSQL: Definition and Introduction – Understanding the Storage Architecture: Working with Column-Oriented Databases, Hbase Distributed Storage Architecture – Document store internals – understanding Key/Value Stores in Memcached and Redis – Performing CRUD operations: Creating records – Accessing data – Updating and Deleting data – Managing Transactions and Data Integrity: Distributed ACID Systems – Upholding CAP.

UNIT V – MongoDB Basics: Data Modeling, CRUD Operations, and Querying
Getting started: Documents – Collections – Databases – Getting and starting MongoDB – Introduction to the MongoDB Shell – Data types – using the MongoDB Shell. Creating , updating, deleting documents: Inserting and Saving documents – Removing documents – Updating documents. Querying: Introduction to Find – Query Criteria – Types Specific Queries.

Textbooks:

1. Silberschatz, Korth, and Sudarshan, Database System Concepts, McGraw-Hill Education, 7th Edition, 2020. (Unit I, II & III)
2. Shashank Tiwari, Professional NoSQL, Wrox Press, Wiley, 2011. (Unit IV)
3. Kristina Chodorow, MongoDB: The Definitive Guide, O'Reilly Media, 3rd Edition, 2019. (Unit V)

Reference Books:

1. Ramez Elmasri & Shamkant B. Navathe, Fundamentals of Database Systems, Pearson Education, 8th Edition, 2022.
2. Raghu Ramakrishnan & Johannes Gehrke, Database Management Systems, McGraw-Hill Education, 3rd Edition, 2018.
3. Rick F. van der Lans, Introduction to SQL, Addison-Wesley, 4th Edition, 2017.
4. Pramod J. Sadalage & Martin Fowler, NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence, Addison-Wesley, 2013.
5. Amit Phaltankar, Juned Ahsan, Michael Harrison, and Liviu Nedov, MongoDB Fundamentals. Packt Publications – 2020.

Discipline Centric Core Course - IV

24UCA504: Data Communication and Networking

Credits: 4 **Internal Marks: 25**

Contact Hours: 4 **External Marks: 75**

Course Outcomes:

Unit No.	CO. No.	Upon completion of this course, the students will be able to	Cognitive Level
Unit I	CO1	Understand basic concepts of data communication, network types, protocols, transmission methods, and media.	K1 – K3
Unit II	CO2	Explain data link control, framing, error control, media access methods, and link layer addressing.	K1 – K3
Unit III	CO3	Illustrate and apply LAN, WAN, wireless technologies, and virtual LAN concepts.	K1 – K4
Unit IV	CO4	Analyze network layer services, IPv4 and IPv6 addressing, routing algorithms, and packet switching.	K1 – K4
Unit V	CO5	Apply transport layer protocols, TCP/UDP, client-server models, and standard network applications.	K1 – K4

Syllabus

Unit I – Fundamentals of Data Communication

Data communication concepts and components – Networks – Network types – Protocol Layering – TCP / IP Protocol Suite – The OSI Model – Signals – Signal Impairment - Digital Transmission – Analog Transmission – Multiplexing – Transmission media.

Unit II – Data Link Layer

Introduction – Data-link control – Framing - Error Control – Media access protocols – Random Access - – Link layer addressing.

Unit III – LAN, WAN and Network Technologies

Local Area Network: Ethernet - WIFI, IEEE 802.11 Project – Bluetooth. Wide Area Network: Telephone Networks - Cable Networks - Cellular Telephony - Satellite Network. Connecting Devices and Virtual LANs: Connecting Devices - Virtual LANS.

Unit IV - Network Layer

Network Layer: Services – Packet switching – Performance – Internet Protocol Version4– IPv4 Addressing – Main and Auxiliary Protocols – Next generation IPv6 – IPv6 Addressing – The IPv6 Protocol - Transmission from IPv4 to IPv6 – Routing of Packets: Introduction - Routing algorithms – Unicast Routing Protocols - Multicast routing.

Unit V – Transport and Application Layers

Transport layer services – Transport-layer protocols – User datagram protocols – Transmission Control Protocol – Application layer: Introduction – Client/Server Paradigm – Standard Applications.

Textbook:

1. Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, McGraw Hill, 2023.

Reference Books:

1. William Stallings – *Data and Computer Communications*, 11th Edition, Pearson, 2022.
2. James F. Kurose & Keith W. Ross – *Computer Networking: A Top-Down Approach*, 8th Edition, Pearson, 2023.
3. Douglas Comer – *Internetworking with TCP/IP*, 7th Edition, Pearson, 2020.
4. Andrew S. Tanenbaum & David J. Wetherall – *Computer Networks*, 6th Edition, Pearson, 2021.

Discipline Centric Core Course – V

24UCA505: Version Control Systems

Credits: 4 **Internal Marks: 25**

Contact Hours: 4 **External Marks: 75**

Course Outcomes:

Unit No.	CO. No.	Upon completion of this course, the students will be able to	Cognitive Level
Unit I	CO1	Explain the basic concepts, components, characteristics, repositories, and internal structure of Git.	K1 – K4
Unit II	CO2	Apply branching and commit operations to manage version history in Git.	K1 – K3
Unit III	CO3	Use Git commands to manage files, track changes, and analyze differences between file versions.	K1 – K4
Unit IV	CO4	Analyze and resolve merge conflicts and modify commit history using advanced Git operations.	K2 – K5
Unit V	CO5	Evaluate and manage remote repositories and collaborate on projects using GitHub.	K2 – K5

Syllabus:

Unit I – Introduction and Foundational Concepts

Introduction to Git – Git components – Git characteristics – Git command line – Quick introduction to using Git –
Repositories – Visualizing the Git object store – Git internals and concepts at work

Unit II – Branches and Commits

Branches – Motivation for using branches – Branching guidelines – Managing branches – Commits – Atomic changesets – Identifying commits – Commit history

Unit III – File Management and Differences

File management and index – Using git add – Using git commit – Using git rm – Using git mv – Tracking renames
– .gitignore file – Diffs – Forms of git diff – Diff output – Commit ranges – Path limiting.

Unit IV – Merging and Intermediate Operations

Merges – Merge examples – Merge conflicts – Merge strategies – How Git handles merges – Finding commits – Using git bisect – Using git blame – Using pickaxe – Altering commits – Using git revert – Using git reset – Using git cherry-pick – Rebasing commits – Stash – Reflog.

Unit V – Remote Repositories and GitHub

Remote repositories – Repository concepts – Remote workflow – Remote configuration – Tracking branches –
Adding and deleting remote branches – Bare repositories and git push – Repository publishing – Repository structure
– Working with multiple repositories – Git and GitHub – History of Git – Installing Git.

Textbook:

1. Jon Loeliger & Matthew McCullough, Version Control with Git, 3rd Edition, O'Reilly Media, 2022,

Reference Books

1. Tsitoara, Mariot. Beginning Git and GitHub: Version Control, Project Management and Teamwork for the New Developer, 2nd Edition, Apress, Berkeley, CA, 2024.
2. Scott Chacon and Ben Straub, Pro Git, 2nd Edition, Apress, 2025.
3. Srinivasan, Sankar. Git Basics and Version Control: Coder's Companion, 1st Edition, Sankar Srinivasan,2023.

Discipline Centric Core Laboratory - V
24UCA553: Version Control Systems Lab

Credits: 2 **Internal Marks: 40**
Contact Hours: 4 **External Marks: 60**

Course Outcomes:

CO. No.	Upon completion of this course, the students will be able to	Cognitive Level
CO1	Create and manage local repositories, add files, and commit changes using GitHub Desktop	K3
CO2	Track file changes, view commit history, and compare file differences	K2
CO3	Manage file versions by staging files, using .gitignore, and restoring previous versions	K3
CO4	Create branches, merge branches, and resolve merge conflicts	K3
CO5	Work with remote repositories by cloning, pushing, pulling, fetching, and collaborating using fork and pull request on GitHub	K4

Syllabus:

1. Create a new local repository using GitHub Desktop, add files, and commit changes.
2. Demonstrate how to view changes, commit history, and file differences using GitHub Desktop.
3. Demonstrate how to stage files, commit multiple versions, and track file modifications.
4. Create and configure a .gitignore file and demonstrate ignoring files.
5. Demonstrate how to discard changes, revert commits, and restore previous versions.
6. Create branches, switch between them, and work on multiple branches.
7. Merge two branches and resolve merge conflicts using GitHub Desktop.
8. Create and publish a repository to GitHub, clone it, and push changes.
9. Demonstrate how to fetch, pull, and synchronize changes between local and remote repositories.
10. Fork a repository, make changes, and create a pull request.

Software Requirements

1. Operating System – Windows 10 / Windows 11
2. Version Control Software – GitHub Desktop
3. Remote Repository Platform – GitHub
4. Code Editor / IDE – Visual Studio Code / Notepad++

Discipline Centric Core Course - VI
24UCA506: Software Project Development

Credits: 4 **Internal Marks: 25**
Contact Hours: 4 **External Marks: 75**

Course Outcomes:

Unit No.	CO. No.	Upon completion of this course, the students will be able to	Cognitive Level
Unit I	CO1	Explain software engineering concepts and different software process models.	K1 – K4
Unit II	CO2	Apply requirements engineering techniques for gathering and analyzing software requirements.	K1 – K4
Unit III	CO3	Design software architecture, components, and user interfaces.	K1 – K4
Unit IV	CO4	Apply software testing methods and quality assurance techniques.	K2 – K4
Unit V	CO5	Apply project management techniques including estimation, metrics, and risk management.	K2 – K4

Syllabus:

Unit I - Introduction to Software Engineering and Process Models
Software Engineering: Defining the Discipline – The Software Process – Software Process Structure: A Generic Process Model – Defining a Framework Activity – Identifying a Task Set – Process Patterns – Process Assessment and Improvement – Process Models: Prescriptive Process Models – Specialized Process Models – The Unified Process – Personal and Team Process Models – Agile Development.

Unit II – Requirements Engineering and Modeling
Understanding Requirements: Establishing the Groundwork – Eliciting Requirements – Developing Use Cases – Building the Analysis Model – Negotiating Requirements – Requirements Monitoring – Validating Requirements – Requirements Modeling: Scenario-based Methods – Class-based Methods.

Unit III – Software Design
Design Concepts: Design Process – Design Concepts – Design Model – Architectural Design: Software Architecture – Architectural Styles – Architectural Design – Component-level Design: Designing Class-based Components – Conducting Component-level Design – Component-based Development – User Interface Design: The Golden Rules – User Interface Analysis and Design – Interface Analysis – Interface Design Steps.

Unit IV – Quality Assurance and Testing
Quality Concepts: The Software Quality Dilemma – Achieving Software Quality – Software Quality Assurance: Elements of Software Quality Assurance – SQA Process and Product Characteristics – SQA Tasks, Goals and Metrics – Statistical Software Quality Assurance – Software Reliability – Software Testing Strategies: Verification and Validation – Unit Testing – Integration Testing – Validation Testing – System Testing – White Box Testing – Basis Path Testing – Control Structure Testing – Black Box Testing.

Unit V – Software Project Management
Software Configuration Management – Process and Project Metrics: Software Measurement – Metrics for Software Quality – Estimation for Software Projects: The Project Planning Process – Software Scope and Feasibility – Software Project Estimation – Empirical Estimation Model – Risk Management: Risk Identification – Risk Projection – Risk Refinement – Risk Mitigation, Monitoring and Management.

Textbook:
1. Roger S. Pressman & Bruce R. Maxim, Software Engineering, 8th Edition, McGraw Hill Education, 2022.

Reference Books:

1. Ian Sommerville, Software Engineering, 10th Edition, Pearson Education, 2021.
2. Bob Hughes & Mike Cotterell , Software Project Management, 6th Edition, McGraw Hill, 2023.
3. Pankaj Jalote , Software Project Management in Practice, Pearson, 2022.
4. K.K. Aggarwal & Yogesh Singh, Software Engineering, 4th Edition, New Age International, 2024.
5. Mall, R., Fundamentals of Software Engineering, 5th Edition, PHI Learning, 2018.

Discipline Centric Elective Courses

Track1: Higher Studies

24UCA701: Distributed Technologies

Credits: 4 **Internal Marks: 25**

Contact Hours: 4 **External Marks: 75**

Course Outcomes:

Unit No.	CO. No.	Upon completion of this course, the students will be able to	Cognitive Level
Unit - I	CO1	Explain the concepts of distributed systems and remote invocation Methods	K1 – K3
Unit - II	CO2	Apply knowledge of distributed objects, components, and web services to design simple distributed applications.	K1 – K4
Unit - III	CO3	Develop basic web applications using ASP.NET Core MVC and the MVC pattern.	K1 – K4
Unit - IV	CO4	Use advanced C# features and Razor syntax to build dynamic and efficient web pages.	K2 – K4
Unit - V	CO5	Analyze and manage software development processes using tools and project management techniques.	K2 – K4

Unit I – Distributed Systems and Remote Invocation

Introduction – Example s of Distributed Systems – Trends in Distributed Systems – Focus on Resource Sharing – Challenges – Case Study: The World Wide Web. **Remote Invocation:** Introduction – Request-reply Protocols – Remote Procedure Call – Remote Method Invocation – Case Study: Java RMI. **Distributed Objects and Components:** Introduction – Distributed Objects – Case Study: CORBA – From Objects to Components – Case Study: Enterprise Java Beans and Fractal.

Unit II – Introduction to ASP.NET Core MVC

Understanding the History of ASP.NET Core MVC - Understanding ASP.NET Core - Installing Visual Studio - Installing the .NET Core 2.0 SDK - Creating a New ASP.NET Core MVC Project - Rendering Web Pages - Creating a Simple Data-Entry Application - Understanding the MVC Pattern - Comparing MVC to Other Patterns - Understanding ASP.NET Core MVC Projects.

Unit III - Advanced C# Features and Razor Syntax

Essential C# Features: Preparing the Example Project - Using the Null Conditional Operator - Using Automatically Implemented Properties- Using String Interpolation - Using Object and Collection Initializers - Pattern Matching - Using Extension Methods - Using Lambda Expressions. Working with Razor: Working with Layouts - Using Razor Expressions.

Unit IV - Development Environment Setup and Project Management

Working with Visual Studio - Preparing the Example Project - Managing Software Packages - Understanding Iterative Development - Preparing JavaScript and CSS for Deployment. Working with Visual Studio Code: Setting Up the Development Environment - Creating an ASP.NET Core Project - Preparing the Project with Visual Studio Code.

Unit V- Web Services and Related Technologies

Web Services: Introduction – Web services – Service description and IDL for web services – A directory

service for use with web services – XML Security – Coordination of web services – Applications of web services. Types of Web Services- Technologies- XML- WSDL- UDDI.

Textbooks:

1. George Coulouris, Jean Dollimore, Tim Kindberg & Gordon Blair, Distributed Systems Concepts and Design, 5th Edition, Pearson Education, 2021. (Unit I & V)
2. Adam Freeman. *Pro ASP.NET Core MVC 2, 7th Edition,* Apress, 2017. (Unit I,III&IV)

Discipline Centric Elective Laboratory

24UCA751: Distributed Technologies Lab

Credits: 2 **Internal Marks: 40**

Contact Hours: 4 **External Marks: 60**

Course Outcomes:

CO No.	Upon completion of this course, the students will be able to	Cognitive Level
CO1	Demonstrate the use of RMI to establish communication between client and server in distributed applications.	K2 – K4
CO2	Develop interactive web applications using JSP and JavaBeans.	K3
CO3	Exam ine XML data structures and implement DOM parsing for data manipulation.	K2, K3
CO4	Implement database operations for managing and displaying records in web applications.	K2 – K4
CO5	Evaluate web service and JSON techniques for exchanging information in distributed environments.	K4

List of Programs:

Basic Programs

1. Write a program to display Hello World in ASP.NET
2. Create a web page to display current date and time
3. Design a page using Label, TextBox, and Button controls
4. Create a program to change label text on button click

Arithmetic Programs

5. Write a program to perform addition of two numbers
6. Create a program to perform basic arithmetic operations (Add, Subtract, Multiply, Divide)
7. Develop a simple calculator application

Form Handling Programs

8. Design a login form with username and password validation
9. Create a registration form with validation controls
10. Implement RequiredFieldValidator, RangeValidator, CompareValidator

Controls-Based Programs

11. Create a program using DropDownList to display selected item
12. Develop a page using RadioButton and CheckBox controls
13. Create a ListBox program to select multiple items
14. Design a File Upload control application

Data Handling Programs

15. Create a program to display data using GridView

Software Requirements:

- Windows Operating System
- Microsoft Visual Studio
- ASP.NET
- Microsoft SQL Server / MySQL
- Internet Information Services (IIS)
- Google Chrome / Microsoft Edge

24UCA705: Software Testing

Credits: 4 **Internal Marks: 25**
Contact Hours: 4 **External Marks: 75**

Course Outcomes:

Unit No.	CO. No.	Upon completion of this course, the students will be able to	Cognitive Level
Unit I	CO1	Identify and design test cases using boundary value testing techniques.	K1 – K4
Unit II	CO2	Design test cases using equivalence class and decision table testing methods.	K1 – K4
Unit III	CO3	Apply structural, data flow, and integration testing techniques to software programs.	K1 – K4
Unit IV	CO4	Perform system testing and object-oriented testing for software systems.	K2 – K4
Unit V	CO5	Apply test management, software metrics, and automation tools in software testing.	K1 – K4

Syllabus:

Unit I – Basics of Software Testing and Boundary Value Testing

A Perspective on Testing: Basic Definition - Test Cases - Venn Diagram - Identifying Test Cases - Fault Taxonomies
- Levels of Testing – Examples - Boundary Value Testing: Normal Boundary Value Testing - Robust Boundary Value Testing - Worst-Case Boundary Value Testing - Special Value Testing – Examples.

Unit II – Equivalence Class and Decision Table Testing

Equivalence Class Testing: Equivalence Classes - Traditional Equivalence Class Testing - Improved Equivalence Class Testing - Equivalence Class Test Cases. Decision Table–Based Testing: Decision Tables - Decision Table Techniques - Test Cases for the Triangle Problem - Test Cases for the NextDate Function - Test Cases for the Commission Problem.

Unit III - Structural and Integration Testing

Path Testing - Program Graphs - DD-Paths - Test Coverage Metrics - Basis Path Testing. Data Flow Testing: Define/Use Testing - Slice-Based Testing - Program Slicing Tools. Integration Testing: Decomposition-Based Integration - Call Graph–Based Integration - Path-Based Integration – Example.

Unit IV – System and Object-Oriented Testing

System Testing: Threads - Basis Concepts for Requirements Specification - Model-Based Threads - Use Case–Based Threads – Non-functional System Testing. Object-Oriented Testing: Issues in Testing Object-Oriented Software - Object-Oriented Unit Testing - Object-Oriented Integration Testing - Object-Oriented System Testing.

Unit V – Test Management, Metrics, and Automation

Test Management: Test Organization – Structure of Testing Group – Test Planning –Detailed Test Design and Test Specification. Software Metrics: Need of Software Measurement – Definition and Classification of Software Metrics. Automation and Testing Tools: Need for Automation – Categorization of testing Tools – Selection of Testing Tools

- Cost Incurred in Testing Tools – Guidelines for Automated Testing.

Textbook:

1. Paul C. Jorgensen, "Software Testing: A Craftsman's Approach", 2021, Fifth Edition, Auerbach Publications.(Unit I, II, III & IV)
2. Naresh Chauhan, *Software Testing: Principles and Practices*, Oxford University Press, 2020. (Unit V).

Reference Books:

1. Dorothy Graham, Foundations of Software Testing, 2020, Fourth Edition, Cengage Publication.
2. Bernard Homès, Fundamentals of Software Testing, ISTE Ltd and John Wiley & Sons, Inc, 2024.
3. Sandeep Desai and Srivastava Abhishek, Software Testing: A Practical Approach, 2nd Edition, PHI Learning Publication, 2016.

24UCA753: Software Testing Tools Lab

Credits: 2 **Internal Marks: 40**
Contact Hours: 4 **External Marks: 60**

Course Outcomes:

CO. No.	Upon completion of this course, the students will be able to	Cognitive Level
CO1	Understand basic software testing concepts, test cases, and test plans	K1, K2
CO2	Apply manual and automated testing to design and run test cases.	K3
CO3	Perform different types of testing using tools.	K3, K4
CO4	Create and run test scripts for web and API testing using tools	K3
CO5	Find bugs, report them, and track their status using bug tracking tools.	K4

Syllabus:

1. Design the test cases for any application using manual testing
2. Create test plan for any applications
3. Perform Regression Testing using RFT tool.
4. Perform Unit Testing using JUnit testing tool.
5. Perform Integration Testing for a sample application and document the test results.
6. Design and execute test cases for a web application using Selenium WebDriver.
7. Perform load and security testing using Selenium Automation Testing tool.
8. Perform performance testing using Apache JMeter testing tool.
9. Develop and execute test scripts for API testing using Postman or REST Assured.
10. Identify bugs in a given application, report them using a bug tracking tool, and track their status.

Software and Tools Requirements:

- Operating System: Windows
- Programming Language: Java
- Testing Tools: JUnit ,Selenium WebDriver ,Apache JMeter, IBM Rational Functional Tester (RFT) ,Postman / REST Assured
- IDE (Development Environment): Eclipse / IntelliJ IDEA

24UCA707: Information Security

Credits: 4 **Internal Marks: 25**
Contact Hours: 4 **External Marks: 75**

Course Outcomes:

Unit No.	CO. No.	Upon completion of this course, the students will be able to	Cognitive Level
Unit I	CO1	Explain the fundamentals of information security and security models.	K1 – K2
Unit II	CO2	Identify and analyze threats, attacks, and vulnerabilities in information systems.	K1 – K4
Unit III	CO3	Assess professional responsibilities and ethical practices in information security.	K1 – K4
Unit IV	CO4	Apply risk management and risk assessment techniques.	K1 – K4
Unit V	CO5	Analyze the effectiveness of access control, firewalls, VPNs, and IDS in securing information systems.	K1 – K4

Syllabus:

Unit I - Introduction to Information Security
Introduction – The History of Information Security – Security – CNSS Security Model – Components of an Information System – Approaches to Information Security Implementation – Security in the Systems Development Life Cycle.

Unit II - The Need for Security
Introduction – Threats and Attacks – Intellectual Property – Human Error or Failure – Information Extortion –
Software Attacks – Technical Hardware and Software Failures - Technological Obsolescence.

Unit III - Legal, Ethical, and Professional Issues
Law and Ethics in Information Security - International Laws and Legal Bodies - Ethics and Information Security - Codes of Ethics of Professional Organizations.

Unit IV - Risk Management
An overview of Risk Management - Risk Identification - Risk Assessment - Risk Control - Quantitative Versus Qualitative Risk Management Practices**.**

Unit V - Security Technologies
Access Controls, Firewalls, and VPNs: Introduction - Access Control – Firewalls - Protecting Remote Connections
- Intrusion Detection and Prevention Systems - Honeypots, Honeynets, and Padded Cell Systems - Scanning and Analysis Tools.

Textbook:
1. Michael E. Whitman & Herbert J. Mattord. Principles of Information Security, 7th Edition. Cengage Learning, 2022.

Reference Books
1. William Stallings. (. Cryptography and Network Security: Principles and Practice (8th Edition). Pearson Education, 2023.
2. David Alexander, Amanda Finch, & Andy Taylor. (2020). Information Security Management Principles (3rd Edition). BCS, The Chartered Institute for IT.
3. Charles P. Pfleeger, Shari Lawrence Pfleeger, & Jonathan Margulies. (Security in Computing). Pearson.

24UCA754: Information Security Lab

Credits: 2 **Internal Marks: 40**
Contact Hours: 4 **External Marks: 60**

Course Outcomes:

CO. No.	Upon completion of this course, the students will be able to	Cognitive Level
CO1	Create secure login systems and validate user input.	K3
CO2	Protect data using encryption, hashing, and strong passwords.	K3, K6
CO3	Build simple network security tools like port scanners and firewalls.	K3, K4, K6
CO4	Detect intrusions and phishing attempts to prevent security threats.	K3, K4
CO5	Implement secure file transfer and digital signature techniques.	K3, K6

Syllabus:

1. Design and implement a secure user authentication system that validates usernames and passwords.
2. Develop a program to check file access permissions and allow access only for authorized users.
3. Design a program to validate user input and restrict unsafe or malicious entries.
4. Write a program to implement Caesar cipher encryption and decryption techniques.
5. Implement a program to perform message encryption and decryption using substitution cipher logic.
6. Develop a password strength checker program to ensure strong and secure passwords.
7. Write a program to generate and verify password hashes using the hashlib library.
8. Create a simple port scanning tool to detect open and closed ports of a given IP address.
9. Develop a basic network packet capture simulation using the scapy module or mock data.
10. Implement a simple firewall rule simulator that blocks or allows specific ports.
11. Write a program to store encrypted passwords and retrieve them securely when required.
12. Develop a digital signature simulation program using hashing algorithms.
13. Design a simple intrusion detection program that monitors repeated failed login attempts.
14. Implement a basic secure file transfer simulation using socket programming and encryption techniques.
15. Write a program to detect phishing emails by checking suspicious keywords or links in a message.

Software Requirements:

- Operating System: Windows
- Programming Environment: Python 3.10
- IDE / Code Editor: Visual Studio Code, Jupyter Notebook
- Networking & Security Tools: Wireshark, Nmap

Ability Enhancement Course Technical Skill
24UCA381: AI Interfaces

Credits: 2 **Internal Marks: 40**
Contact Hours: 4 **External Marks: 60**

Course Outcomes:

CO. No.	Upon completion of this course, the students will be able to	Cognitive Level
CO1	Understand the basics of AI interfaces and prompt usage.	K1, K2
CO2	Use AI tools like ChatGPT, Gemini, and Copilot for text and code generation.	K3
CO3	Create visual content and data insights using tools like Canva, Power BI, and Tableau.	K3
CO4	Develop simple AI models and applications using tools like Teachable Machine and LangChain.	K3, K4
CO5	Compare and evaluate outputs from different AI tools and build a mini project.	K4, K5

Syllabus:

1. Create an account and explore the interface of ChatGPT and generate a technical paragraph using a prompt.
2. Use Google Gemini to summarize a given article and extract key points.
3. Use Microsoft Copilot to generate a Python program and explain the code.
4. Use Canva to create a poster or presentation using AI tools.
5. Use DALL·E to generate an image using a text prompt.
6. Use GitHub Copilot to generate a simple program.
7. Use Microsoft Power BI to create a simple dashboard using dataset.
8. Use Tableau to create a data visualization report.
9. Use Teachable Machine to build a simple image classification model.
10. Use Runway ML to generate a short AI video or image.
11. Use LangChain to create a simple chatbot interface.
12. Compare the outputs of ChatGPT, Gemini, and Copilot for the same prompt and analyze the results.
13. Write effective prompts for text generation, summarization, and translation.
14. Generate interview questions using AI tools.
15. Create a mini project using any AI interface tool.

Software Requirements:

- Operating System: Windows 10 / Linux / macOS
- Web Browser: Google Chrome (latest version)
- Programming Language: Python 3.x
- IDE: VS Code / Jupyter Notebook
- Basic Tools Access: Web-based AI tools, Canva, Power BI, Tableau Public