Post Graduate Diploma in Computer Applications

Rules-Regulations & Curriculum

(w.e.f. 2025-26)

REGISTRAR
Sai Tirupati University
Udaipur (Raj.)

Dr. Director

PIMS Institute of Computer Sciences

(A Constituent Unit of Sai Tirupati University, Udaipur)

PROGRAM OUTCOMES (POs)

PO01: Domain Knowledge
Apply the knowledge of computer applications, programming, databases, and networking to solve real-world IT problems effectively.

PO02: Problem Analysis
Identify, analyse, and solve computing problems using logical reasoning and appropriate tools and techniques.

PO03: Design/Development of Solutions
Design and develop software applications and systems that meet user requirements with consideration for efficiency, security, and usability.

PO04: Modern Tool Usage
Use modern computing tools, programming languages, and technologies to develop and implement solutions.

PO05: Professional Ethics
Understand and apply ethical principles and professional responsibilities in the field of information technology.

PO06: Individual and Team Work
Work effectively as an individual and as a team member in multidisciplinary environments.

PO07: Communication Skills
Communicate effectively through technical reports, documentation, and presentations.

PO08: Project Management & IT Practices
Apply knowledge of project management principles and IT practices in software development and implementation.

PO09: Life-long Learning
Recognize the need for continuous learning and adapt to emerging technologies in the field of computer applications.

PROGRAM EDUCATIONAL OUTCOMES (PEOs)

PEO 1: Professional Competency
Develop strong technical and programming skills to establish successful careers in the IT industry or related fields.

PEO 2: Lifelong Learning & Adaptability
Engage in continuous learning and adapt to emerging technologies in computer applications and related domains.

PEO 3: Problem Solving & Innovation
Apply analytical thinking and problem-solving skills to design efficient IT solutions and contribute to innovation.

PEO 4: Ethics, Communication & Social Responsibility
Demonstrate professional ethics, effective communication, teamwork, and a sense of responsibility towards society and sustainable development.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO 1: Programming & Software Development
Develop and implement software applications using programming languages such as C, C++, and Python with structured and object-oriented approaches.

PSO 2: Database & Data Management
Apply database concepts and SQL techniques to design, manage, and manipulate data efficiently for real-world applications.

PSO 3: Web Development & Networking
Design and develop basic web applications and understand data communication and networking concepts for effective information exchange.

PSO 4: Practical Skills & Industry Readiness
Demonstrate practical knowledge, problem-solving ability, and professional skills through lab work, projects, and real-world applications in the IT field.

ADMISSION

Eligibility Criteria

No candidate shall be allowed to be admitted to the **Post Graduate Diploma in Computer Applications (PGDCA)** course unless:

- He/She has passed a **Bachelor's Degree (Graduation)** in any discipline from a recognized university.
- Admission will be granted as per university norms and guidelines.

DURATION OF PROGRAM

The duration of the PGDCA program will be One Year**.**

ATTENDANCE REQUIREMENT

- Minimum **75% attendance** in theory and practical separately is mandatory.
- Students failing to meet attendance criteria will **not be eligible** to appear in the main examination.

SCHEME OF EXAMINATION

a. Examinations will be conducted through **theory papers, practicals, and viva-voce**.
Students must pass **separately in theory and practical**, both in internal and external exams.
Medium of examination: **English/Hindi**.

b. No candidate will be allowed to appear unless he/she has **75% attendance** in each subject.

c. There will be **two examinations** (Internal & External) in each paper.
Students failing or absent can appear in the **next examination cycle**.

d. A candidate must secure:

- **Minimum 40% marks** in each internal and external exam
- **50% aggregate marks** to pass

DIVISION / CLASS

- **First Division** – 60% and above
- **Second Division** – 50% to below 60%

★ *Distinction:*
Candidates securing **75% or above in a subject (first attempt)** will be awarded **Distinction (D)**.

PAPER PATTERN

Internal Examination

- Maximum Marks: **30**
- Minimum Passing Marks: **12**

External Examination

- Maximum Marks: **70**
- Minimum Passing Marks: **28**

Question Paper Pattern

Section A

- 10 Questions (1 mark each)
- All compulsory
- (10 × 1 = 10 marks)

Section B

- 10 Questions (2 from each unit)
- Attempt 1 from each unit
- (5 × 6 = 30 marks)

Section C

- 4 Questions
- Attempt any 2

(2 × 15 = 30 marks)

L- Lecture | T-Theory | P-Practical | C-Credit |

IA- Internal Assessment | EA- External Assessment

SEC- Skill Enhancement Course | DCC- Discipline Core Course | AEC- Ability Enhancement Course

Program Scheme

Post Graduate Diploma in Computer Applications

S.No.	Course Code	Course Name	Category	L	T	P	Credit	Theory		Tutorial	Practical		Total Marks
								IA	EA		IA	EA	
1	313IT101	Computer Fundamentals& Information Technology	SEC	3	0	0	3	30	70	-	-	-	100
2	313OS102	Operating System	DCC	3	0	0	3	30	70	-	-	-	100
3	313PC103	Programming in C & C++	DCC	3	0	0	3	30	70	-	-	-	100
4	313PC104	Programming in C & C++ Lab	SEC	0	0	3	3	-	-	-	30	70	100
5	313DB105	Database Management System	DCC	3	0	0	3	30	70	-	-	-	100
6	313DB106	Database Management System Lab	SEC	0	0	3	3	-	-	-	30	70	100
7	313PY107	Introduction to Python	SEC	3	0	0	3	30	70	-	-	-	100
8	313PY108	Introduction to Python Lab	SEC	0	0	3	3	-	-	-	30	70	100
9	313DC109	Data Communication & Networking	DCC	3	0	0	3	30	70	-	-	-	100
10	313WD110	Web Designing &Technologies Lab	SEC	0	0	3	3	-	-	-	30	70	100
		Total					30						1000

Course Code	Course Name	L	T	P	Credit
313IT101	Computer Fundamentals& Information Technology	3	0	0	3

Course Objectives: Students will understand the basic parts, types, and uses of computers and how computer programming works using flowcharts and algorithms. Students will learn about computer software, hardware, and different types of programming languages and their uses.

Course Outcomes(s): After completion of this course, students shall be able to:

1. Explain the fundamental concepts, evolution, and applications of computers across various fields.
2. Develop problem-solving skills using algorithms, flowcharts, and pseudocode.
3. Demonstrate knowledge of computer hardware, software, and functional components.
4. Apply number systems and set theory for data representation and operations.
5. Use MS Office tools (Word, Excel, PowerPoint) for professional documentation and data analysis.

Unit 1:

Introduction to Computers: Characteristics of computers, Evolution of computers, generation of computers, classification of computers, applications of computers. Computer Program: Introduction, developing a program, algorithm, flowchart, pseudo code. Computer Languages: Introduction, classification of programming languages, generations of programming languages, features of a good programming language. Computer Software: Software definition, relationship between software and hardware, software categories, system software, application software, utility software.

Unit 2:

Input and Output Devices: Keyboard, pointing devices, speech recognition, digital camera, scanners, optical scanners. Classification of output devices, Hard copy output devices- printers, plotters, computer output microfilm (COM), Classification of output devices, Soft copy output devices- monitors, audio output, projectors, and terminals. Computer System: Central processing unit (CPU), Memory, instruction format, instruction set.

Unit 3:

Primary and Secondary Memory: Memory hierarchy, Random access memory (RAM), types of RAM, Read only memory (ROM), types of ROM. Classification of secondary storage devices, magnetic tape, magnetic disk, optical disk. Number Systems: Introduction to number system, Binary, Octal, Hexadecimal, conversion between number bases, Alphanumeric- EBCDIC and ASCII, Sets Theory, Types of Sets, Multi Sets, Operations on Sets.

Unit 4:

Office Management Tools: MS-Word: Creating Saving documents, Entering, Editing, Page formatting, Finding and replacing text, Spell checking and Grammar checking, Indexing, Columns, Tables and feature there in, Inserting (Objects, picture, files etc.), Using Graphics, using Mail Merge, using Word Art, customizing MS Word. MS Excel: Spreadsheet terminology, organization of the worksheet area, editing cells using commands and functions, formatting worksheet, creating & editing charts, naming range and using statistical, mathematical and financial functions, multiple worksheets and Macros, working with objects, Worksheet printing options.

Unit 5:

MS Power Point: Anatomy of a Power Point Presentation, Creating and Viewing a presentation, Managing Slide Shows, using hyperlinks, advanced navigation with action setting and action buttons, organizing formats with Master Slides, adding graphics, multimedia and special effects, creating presentations for the web.

Text/Reference Books

1. *P.K. Sinha, Fundamentals of Computers, BPB Publications, 2021*
2. *R. Thareja, Fundamentals of Computers, Oxford University Press, 2020*
3. *ITL Education Solutions, Introduction to Information Technology, Pearson Education, 2022*

Course Code	Course Name	L	T	P	Credit
313OS102	Operating System	3	0	0	3

Course Objectives: To provide a solid understanding of the fundamental concepts and functions of operating systems including process, memory, file, and device management. To analyse different scheduling, concurrency, and deadlock handling mechanisms used in operating systems.

Course Outcomes: After successful completion of this course, students shall be able to:

1. Understand OS fundamentals including types, services, and system calls.
2. Analyse process management with scheduling algorithms and state transitions.
3. Implement memory management techniques like paging, segmentation, and virtual memory.
4. Resolve deadlocks using prevention, avoidance, detection, and recovery methods.
5. 5.Manage process concurrency with synchronization tools like semaphores and IPC.

Unit 1: Introduction: Definition of Operating System, Types of operating systems: Batch Systems, Multi programming, Multiuser, Multitasking, Time–sharing, Spooling, Parallel, Distributed and Real-time systems, Operating System Concepts, Operating System Services, System calls.

Unit 2: Process Management: Process concept, Process States, Representation of process (PCB), Process Scheduling, CPU Scheduling: Scheduling Criteria, Scheduling Algorithms, Algorithm evaluation.

Unit 3: Memory Management: Contiguous, Non-contiguous, Swapping, Fragmentation, Compaction, Paging, Segmentation, Virtual memory management, Demand paging, Page replacement and Virtual memory concepts, Introduction to Thrashing.

Unit 4: The Deadlock problem: Characterization (Hold and wait, Circular Wait, No Pre-emption, no sharing of resources), Prevention, Avoidance: (RAG and Wait for Graph), Detection and Recovery from Deadlock: (Banking algorithm and detection algorithm), Concept of Fork and Join methods.

Unit 5: Process concurrency: Concept of concurrency, cooperating process, precedence graph, Critical section problem, Mutual exclusion, semaphores, classical process (Reader Writer problem, Consumer producer problem, Dining Philosopher problem), Inter Process Communication.

Textbooks/Reference books:

1. Abraham Silberschatz, Peter B. Galvin, Greg Gagne – Operating System Concepts, Wiley, 2018.
2. Andrew S. Tanenbaum, Herbert Bos – Modern Operating Systems, Pearson Education, 2015.
3. William Stallings – Operating Systems: Internals and Design Principles, Pearson Education, 2018.
4.D.M. Dhamdhere – Operating Systems: A Concept-based Approach, McGraw Hill Education, 2012. 5. Achyut S. Godbole, Atul Kahate – Operating Systems, McGraw Hill Education, 2017.

Course Code	Course Name	L	T	P	Credit
313PC103	Programming in C & C++	3	0	0	3

Course Objective: To introduce the fundamentals of **C and C++ programming**, including syntax, data types, control structures, functions, pointers, and file handling. The course aims to develop problem-solving skills and enable students to write efficient, structured, and modular programs using both procedural and object-oriented programming approaches.

Course Outcomes (COs): After completion of this course, students shall be able to:

CO1: Understand the basic concepts of **C programming**, including syntax, data types, operators, and control structures.
CO2: Develop programs using **functions, arrays, pointers, and file handling** in C.
CO3: Understand the concepts of **object-oriented programming (OOP)** in C++ including classes and objects.
CO4: Implement OOP features such as **constructors, inheritance, function overloading, and polymorphism** in C++.
CO5: Apply programming skills to **solve real-world problems** using both C (procedural approach) and C++ (object-oriented approach).

Unit 1: Basics of C Programming: Introduction to C language, history and features of C, structure of a C program, character set, tokens, keywords, identifiers and constants. Data types and variables, declaration and initialization of variables. Operators in C including arithmetic, relational, logical, assignment and conditional operators. Expressions and statements. Library functions. Basic input and output functions such as printf, scanf, getchar and putchar.

Unit 2: Control Structures and Arrays in C: Decision making using if, if-else and switch statements. Looping using while, do-while and for loops. Use of break, continue and goto statements. Concept of arrays, declaration and initialization of one-dimensional and two-dimensional arrays. Strings and basic string operations. Arrays of strings.

Unit 3: Functions, Pointers and Files in C: Concept of functions, function declaration and definition, function prototypes and types of functions. Passing arguments to functions, recursion and passing arrays to functions. Introduction to pointers, pointer declaration and applications, call by reference. Basics of file handling including opening, closing, reading and writing files.

Unit 4: Fundamentals of C++ and OOP: Introduction to Object Oriented Programming, features and advantages of OOP, comparison of procedural and object-oriented programming. Basics of C++ language, tokens, keywords, variables and data types. Input and output using cin and cout. Control statements and loops in C++. Concept of classes and objects, member functions and basic constructors.

Unit 5: C++ Advanced Basics: Concept of inheritance and its types. Function overloading and operator overloading. Introduction to polymorphism and virtual functions. Basic file handling in C++ including opening, closing, reading and writing files.

Textbooks/Reference books:

1. *E. Balagurusamy* – Programming in ANSI C*, McGraw Hill, 2017;*
2. *Yashavant Kanetkar* – Let Us C*, BPB Publications, 2020;*
3. *E. Balagurusamy* – Object Oriented Programming with C++*,*
4. *McGraw Hill, 2019; Herbert Schildt* – C++: The Complete Reference*, McGraw-Hill, 2003.*

Course Code	Course Name	L	T	P	Credit
313PC104	Programming in C & C++ Lab	0	0	3	3

Practical based on the Course **Programming in C and C++ in 313PC103**. This course helps in getting practical exposure to C and C++ languages, which are programming languages that develop the overall understanding and programming skills of students and build a strong foundation in procedural as well as object-oriented programming.

List of Experiments:

1. Write a program to print "Hello World".
2. Program to add two numbers.
3. Program to find area of circle.
4. Program to swap two numbers (without third variable).
5. Program to check even or odd number.
6. Program to find largest of three numbers.
7. Program to check positive, negative or zero.
8. Program to calculate simple interest.
9. Program to convert Celsius to Fahrenheit.
10. Program to find square and cube of a number.
11. Program to print numbers from 1 to N using loop.
12. Program to find sum of first N natural numbers.
13. Program to print multiplication table.
14. Program to find factorial of a number.
15. Program to check prime number.
16. Program to print Fibonacci series.
17. Program to reverse a number.
18. Program to check palindrome number.
19. Program to count digits in a number.
20. Program to find sum of digits.
21. Program to input and display array elements.
22. Program to find largest element in array.
23. Program to find smallest element in array.
24. Program to sort array (ascending order).
25. Program to search element in array (linear search).
26. Program to add two matrices.
27. Program to multiply two matrices.
28. Program to find length of string.
29. Program to reverse a string.
30. Program to check palindrome string.
31. Program using function to find factorial.
32. Program using function to check prime number.
33. Program using recursion for Fibonacci series.
34. Program to demonstrate pointer usage.
35. Program to swap numbers using pointers.
36. Program to demonstrate class and object.

37. Program to implement constructor.
38. Program for function overloading.
39. Program for inheritance.
40. Program to demonstrate polymorphism (virtual function).

Course Code	Course Name	L	T	P	Credit
313DB105	Database Management System	3	0	0	3

Course Objective: This course provides an introduction to the fundamentals of database management systems, data models, entity-relationship modelling, relational model, SQL queries, and the file system structure.

Course Outcomes: After successful completion of this course, students shall be able to:

1. Understand basic concepts and architecture of DBMS and differentiate between file systems and databases.
2. Apply ER modelling techniques and convert ER diagrams to relational tables.
3. Understand and apply relational algebra and relational constraints.
4. Execute SQL queries including joins, subqueries, aggregate functions, and privilege management.
5. Explain file organization techniques and indexing structures including B-trees and hashing.

Unit 1:
Introduction: An overview of database management system, Database System Vs File System, Database system concepts and architecture, data models schema and instances, data independence and database language and their interfaces, Data definition language, DML, Overall Database Structure.
Unit 2:
Data modelling using the Entity Relationship Model: ER model concepts, notation for ER diagram, mapping constraints, keys, Concepts of Super Key, candidate key, primary key, Generalization, aggregation, reduction of an ER diagram to tables, extended ER model.
Unit 3:
Relational data Model and Language: Relational data model concepts, integrity constraints: entity integrity, referential integrity, Keys constraints, Domain constraints, relational algebra.
Unit 4:
SQL: Characteristics of SQL, Advantages of SQL, SQL data types and literals, Insert, update and delete operations, sub queries, Aggregate functions, Joins, Unions, Intersection, Minus operations. Roles and Privileges.
Unit 5:
File and system structure: overall system structure, file organization, logical and physical file organization, sequential and random, hierarchical, inverted, nullity, indexing and hashing, B-tree index files.

Text/Reference Books:

1. *Ramez Elmasri & Shamkant B. Navathe, Fundamentals of Database Systems, Pearson Education, 2020.*
2. *Silberschatz, Korth & Sudarshan, Database System Concepts, McGraw-Hill Education, 2019.*
3. *C.J. Date, An Introduction to Database Systems, Pearson Education, 2018.*
4. *Raghu Ramakrishnan & Johannes Gehrke, Database Management Systems, McGraw-Hill Education, 2021.*

Course Code	Course Name	L	T	P	Credit
313DB106	Database Management System Lab	0	0	3	3

Exercises based on creating table, inserting data into tables, viewing data in the tables, sorting data in table, deleting tuples from table, updating the contents of a table, modifying the structure of table, applying primary key, foreign key and unique key constraints, computations on table data, oracle functions, grouping data from tables, sub queries, Joins.

List of Experiments:

1. Write an SQL query to create a database for student management.
2. Write an SQL query to create a table for students with fields: id, name, course, marks.
3. Write an SQL query to insert records into the student table.
4. Write an SQL query to display all records from the student table.
5. Write an SQL query to display specific columns from a table.
6. Write an SQL query to update student marks using UPDATE command.
7. Write an SQL query to delete a specific record from a table.
8. Write an SQL query to drop a table from the database.
9. Write an SQL query to display students whose marks are greater than 70.
10. Write an SQL query to display students whose names start with letter 'A'.
11. Write an SQL query using AND operator.
12. Write an SQL query using OR operator.
13. Write an SQL query using NOT operator.
14. Write an SQL query using BETWEEN operator.
15. Write an SQL query using IN operator.
16. Write an SQL query using LIKE operator.
17. Write an SQL query to find the total number of students.
18. Write an SQL query to find maximum marks.
19. Write an SQL query to find minimum marks.
20. Write an SQL query to calculate average marks.
21. Write an SQL query to calculate sum of marks.
22. Write an SQL query using GROUP BY clause.
23. Write an SQL query using HAVING clause.
24. Write an SQL query to count students in each course.
25. Create two tables Students and Courses and insert data.
26. Write an SQL query to display data using INNER JOIN.
27. Write an SQL query using LEFT JOIN.
28. Write an SQL query using RIGHT JOIN.
29. Write an SQL query using FULL JOIN (conceptual if DB supports it).
30. Write an SQL query to display student name with course name.
31. Write an SQL query to find students enrolled in a specific course.
32. Write an SQL query to display students with no matching course.
33. Write an SQL query to create a VIEW for student details.
34. Write an SQL query to create an INDEX on student name.

35. Write an SQL query to create a stored procedure.
36. Write an SQL query to create a trigger on student table.
37. Write an SQL query to display top 5 records from a table.
38. Write an SQL query using subquery to find students with marks above average.
39. Write an SQL query to find duplicate records in a table.
40. Write an SQL query to create and enforce primary key and foreign key constraints.

Course Code	Course Name	L	T	P	Credit
313PY107	Introduction to Python	3	0	0	3

Course Objectives: To introduce students to the fundamentals of Python programming, focusing on basic syntax, data types, and control structures. To develop students' skills in using Python for solving computational problems, including the use of functions, loops, and database operations.
Course Outcomes: After successful completion of this course, students shall be able to:

1. Set up Python environment and write basic programs using syntax, variables, and data types.
2. Apply Python operators and manipulate strings using indexing, slicing, and built-in methods.
3. Use Python data structures (lists, tuples, dictionaries) for efficient data storage and retrieval.
4. Implement control flow (conditionals, loops) and functions (including recursion) for program logic.
5. Perform database operations in MySQL using Python for CRUD (Create, Read, Update, Delete) functionality.

Unit 1: **Python Introduction and Setting up the Environment**: Introduction to Programming, Choosing Python, setting up Python environment, Python IDEs, Python Basic Syntax and Data Types, Input/output, Comments, Variables, Data types, Typecasting.

Unit 2: **Operators**: Arithmetic, Assignment, Comparison, Logical, Identity, Membership, Bitwise operators. Strings in Python: Creating, Formatting, Indexing, Slicing, String methods.

Unit 3: **Lists**: Creating, Properties, Indexing, Slicing, Methods, Modifying lists. Tuples: Syntax, Properties, Indexing, Slicing, Methods. Dictionaries: Syntax, Keys/values, Accessing, Methods.

Unit 4: **Conditional Statements**: if, if-else, if-else-else, Loops in Python: while, for, break, continue, List and Dictionary Comprehensions Syntax, uses. Functions in Python: Creating, Calling, Arguments, Variables, Recursion.

Unit 5: **Database Access**: MySQL, CRUD operations.

Textbooks/Reference books:

1. *Dr. Charles R. Severance – Python for Everybody: Exploring Data Using Python 3, CreateSpace Independent Publishing, 2016.*
2. *Mark Lutz – Learning Python, O'Reilly Media, 2013.*
3. *Reema Thareja – Python Programming Using Problem Solving Approach, Oxford University Press, 2017.*
4. *Allen B. Downey – Think Python: How to Think Like a Computer Scientist, O'Reilly Media, 2015.*
5. *Wesley J. Chun – Core Python Programming, Pearson Education, 2012.*

Course Code	Course Name	L	T	P	Credit
313PY108	Introduction to Python Lab	0	0	3	3

List of Experiments:

1. Download and install Python from the official website OR Choose and set up an IDE (like PyCharm, VSCode, or Jupyter Notebook) for Python development.
2. Create a Python script that prints "Hello, World!" to the console. OR Add comments explaining each line of code.
3. Write a Python program that defines variables of different data types (integer, float, string, Boolean) and prints their values and types.
4. Write a Python script that reads user input as a string and converts it into different data types (into, float) for further use.
5. Create a Python program that demonstrates the use of arithmetic, comparison, logical, and bitwise operators with example expressions.
6. Write a Python script that creates a string, formats it using f-strings or format method, and performs operations like indexing, slicing, and using string methods (e.g., upper (), find (), replace ()).
7 Create a Python program that initializes a list with several elements, performs operations such as indexing, slicing, and applying methods like append (), remove (), and sort ().
8 Write a script that defines a tuple with some elements and demonstrates accessing elements, slicing, and using methods (e.g., count (), index ()).
9. Write a Python program that creates a dictionary with key-value pairs, demonstrates accessing values, adding new entries, and updating existing entries.
10. Create a script that takes user input and uses if, if-else, and if-elif-else statements to provide different responses based on the input.
11. Write a Python program that uses both while and for loops to iterate over a range of numbers and a list, and demonstrates the use of break and continue statements.
12. Write a Python script that generates a list of squares of numbers using list comprehension and a dictionary with squares as keys and their roots as values using dictionary comprehension.
13. Create a Python function that takes parameters, performs some calculations, and returns a result. Write code to call this function with different arguments and print the results.
14. Write a recursive function to solve a problem like calculating factorial or generating Fibonacci numbers and test it with different values.
15. Set up a MySQL database and write Python scripts using the MySQL-connector-python package to connect to the database and perform CRUD (Create, Read, Update, Delete) operations on a sample table

Course Code	Course Name	L	T	P	Credit
313DC109	Data Communication & Networking	3	0	0	3

Course Objective: This course introduces students to the fundamentals of computer networks, data communication concepts, network protocols, transmission media, and network security. The course builds a strong foundation for understanding networking architectures and implementation techniques.

Course Outcome(s): After successful completion of this course, students shall be able to:

1. Understand the evolution of computer networks, communication systems, and transmission models.
2. Describe networking architectures including OSI and TCP/IP models and their layers.
3. Apply data communication concepts including signal encoding, modulation, and LAN standards.
4. Explain error detection techniques and basic network security measures.
5. Identify different transmission media, LAN topologies, and access techniques.

Unit 1:

Principles of Data Communication: Evolution of computer networks, General features and tasks of a communication system, Fundamentals of signals, carrier waves, modes of transfer: simplex, half duplex, full duplex, types of networks: LAN, WAN, MAN, SAN, PAN, CAN, VPN, EPN. Introduction to serial communication.

Unit 2:

Networking Architecture: ISO-OSI, IBM SNA architecture –their functions of each layer and implementation. Concepts of circuit switching, packet switching and message switching. Fundamentals of datagrams. Flow and Error Control – Stop and Wait, Sliding Window, Automatic Repeat Request.

Unit 3:

Data communication concepts: Connecting devices, hub, switch, bridge, routers and gateways, Signal encoding and decoding techniques - Amplitude Modulation, Frequency Modulation, Phase Modulation, signal bandwidth requirements, signal formats used in LAN, Network Protocols: LAN cabling standards: IEEE LAN standards.

Unit 4:

Error detection and correction codes: Parity bit, Checksum, Hamming codes, CRC, single error detection and correction. Introduction to Network Security Model, concepts of key, Ceaser cipher, transposition cipher, DES.

Unit 5:

Transmission media: twisted pair, coaxial cable, optical-fiber. LAN topologies: STAR, BUS and RING network, LAN access techniques: ALOHA, CSMA, token ring and token bus. Issues related with network reliability and fault redundant network systems.

Text/Reference Books:

1. *Behrouz A. Forouzan ,Data Communications and Networking, McGraw Hill 2017.*
2. *William Stallings , Data and Computer Communications, Pearson Education, 2013.*
3. *Andrew S. Tanenbaum & David J. Wetherall – Computer Networks, Pearson, 2011.*
4. *Prakash C.Gupta, Data Communication and Computer Networks, PHI Learning Pvt.Ltd., 2013.*

Course Code	Course Name	L	T	P	Credit
313WD110	Web Designing &Technologies Lab	0	0	3	3

Course Objectives: To introduce foundational web technologies such as HTML and CSS for webpage development. The course aims to equip students with practical skills to build and design static, responsive websites using modern tools like WordPress.

Course Outcomes(s): After successful completion of this course, students shall be able to:

1. Understand and apply HTML elements to create structured web pages. Use tags for formatting text, inserting images, and creating lists and links.
2. Design and develop interactive web forms using basic and advanced HTML form elements. Implement internal links and image maps for improved navigation.
3. Apply CSS to control the style, layout, and design of web pages effectively. Differentiate between inline, internal, and external stylesheets.
4. Create responsive and visually appealing websites using WordPress. Manage themes, plugins, and multimedia content through WordPress dashboards.
5. Demonstrate the ability to integrate HTML and CSS to produce well-structured, user-friendly web designs. Develop static websites following standard web design practices.

Experiments related to topics below:

Unit 1: HTML Basics

Introduction to HTML, structure of an HTML document and basic tags. Creating web pages using headings, paragraphs, line breaks and text formatting tags. Working with lists and comments. Creating hyperlinks and inserting images in web pages.

Unit 2: Tables and Forms

Creating tables using rows, columns and table attributes. Designing HTML forms with input fields, text boxes, radio buttons, checkboxes, dropdown lists and buttons. Basic form layout design.

Unit 3: Cascading Style Sheets (CSS)

Introduction to CSS and its types: inline, internal and external CSS. Styling text, fonts, colors and backgrounds. Applying CSS to tables and forms. Designing simple page layouts using CSS and div tags.

Unit 4: JavaScript Basics

Introduction to JavaScript and its use in web pages. Writing simple JavaScript programs using variables and operators. Using alert, prompt and confirm boxes. Basic form validation using JavaScript.

Unit 5: Website Development and Hosting

Developing a small website with multiple pages. Linking pages together and applying consistent styles. Basic idea of domain and web hosting. Uploading website files on a hosting platform.

Text/Reference Books:

1. *Jon Duckett – HTML and CSS: Design and Build Websites, Wiley, 2011.*
2. *Jon Duckett – JavaScript and JQuery, Wiley, 2014.*
3. *Thomas A. Powell – HTML & CSS: The Complete Reference, McGraw Hill, 2017.*
4. *David Flanagan – JavaScript: The Definitive Guide, O'Reilly, 2020.*