Jharkhand University of Technology, Ranchi

Syllabus

of

Bachelor of Computer Applications (BCA)

(With effect from Academic Year 2025-26)

Department of Computer Science

SEMESTER-III

Computer Organization and Architecture

Course Objectives:

The objectives of this course are to

1. Introduce the internal organization and architecture of computer systems
2. Explain CPU structure, instruction execution and control mechanisms
3. Develop understanding of memory hierarchy and I O organization
4. Familiarize students with computer architecture concepts and designs
5. Provide exposure to modern and advanced computer architectures

Course Outcomes:

After completing this course, the student will be able to

CO1: Explain basic concepts of computer organization and functional units

CO2: Describe computer architecture concepts and system design

CO3: Understand CPU organization and instruction execution

CO4: Explain memory organization and I O mechanisms

CO5: Understand modern computer architecture and performance concepts

Week-wise Syllabus Plan

Week	Course Outcomes	Program Outcomes	Topics	Contents
Week 1	CO1	PO1, PO2, PO5, PO12	Computer Fundamentals	• Definition • Main Function Execution of program • Block diagram of Computer • Functional unit and their Functions
Week 2	CO1	PO1, PO2, PO5, PO12	Computer Evolution and Performance	• Evolution of computer systems • Generations of computers • Moore's law • Basic performance measures
Week 3	CO2	PO1, PO2, PO3, PO5, CO10, PO12	Computer Architecture Fundamentals	• Concept of computer architecture • Organization vs architecture • Instruction set architecture • Von Neumann architecture • Harvard architecture
Week 4	CO2	PO1, PO2, PO3, PO5, PO10, PO12	Advanced Computer Architecture Concepts	• RISC and CISC architectures • Superscalar architecture basics • Pipelining concept • Flynn's classification of computers
Week 5	CO3	PO1, PO2, PO3, PO5, CO10, PO12	Basic Computer Organization	• Stored program concept • Instruction codes • Register organization • Types of registers
Week 6	CO3	PO1, PO2, PO3, PO5, CO10, PO12	Instruction Cycle	• Fetch cycle • Decode cycle • Execute cycle • Interrupt cycle
Week 7	CO3	PO1, PO2, PO3, PO5, CO10, PO12	Instruction Formats and Addressing Modes	• Instruction formats • Types of instructions • Addressing modes immediate direct

				indirect register
Week 8	CO3	PO1, PO2, PO3, PO5, CO10, PO12	Central Processing Unit	• Arithmetic Logic Unit • Control Unit • Hardwired control • Microprogrammed control
Week 9	CO4	PO1, PO2, PO3, PO5, PO10, PO12	Memory Organization	• Memory hierarchy • Primary memory • Cache memory basics • Secondary storage overview
Week 10	CO4	PO1, PO2, PO3, PO4, PO5, PO10, PO12	Input Output Organization	• 10 devices • 10 interface • Programmed 10 • Interrupt driven 10 • Direct Memory Access
Week 11	CO4	PO1, PO2, PO3, PO4, PO5, PO10, PO12	System Bus and Peripheral Devices	• System bus structure • Data bus, address bus and control bus • Peripheral devices and controllers
Week 12	CO5	PO1, PO2, PO3, PO4, PO5, PO9, PO10, PO12	Modern Processor Architecture	• Multicore processors • Parallel processing basics • GPU overview • Virtual machines basics
Week 13	CO5	PO1, PO2, PO3, PO4, PO5, PO9, PO10, PO12	Latest Trends in Computer Architecture	• Cloud computing architecture overview • Edge computing basics • Green computing • Emerging hardware trends

Suggested Additional Activities

1. Online Resources and Learning Platforms
   a. GeeksforGeeks - Computer Organization
      https://www.geeksforgeeks.org/computer-organization-and-architecture
      Easy explanations with diagrams and examples.
   b. TutorialsPoint — Computer Organization
      https://www.tutorialspoint.com/computer_organization
      Good for quick revision.
   C. NPTEL Video
      Lectures
      https://npteI.ac.in
      Indian university level lectures on COA.
   d. MIT
      OpenCourseWare
      https://ocw.mit.edu
      For advanced understanding of computer architecture.
   e. YouTube
      Channels Gate
      Smashers Neso
      Academy Jenny's
      Lectures
      Helpful for visual learning.
   f. Software and Learning Tools (Optional
      Exposure) CPU Simulator tools
      Online architecture simulators
      Performance analysis videos
2. Reference Books
   a. Computer Organization and Architecture — William
      Stallings Very clear explanation of architecture concepts.
   b. Computer System Architecture — M. Morris Mano
      Suitable for understanding organization concepts.
   C. Computer Organization — Carl Hamacher
      Good coverage of CPU and memory topics.
   d. Structured Computer Organization — Andrew S. Tanenbaum
      Explains architecture from basic to advanced level.

Data Communication and Computer Networks

Course Objectives

The objectives of this course are to

1. Introduce basic concepts of data communication and computer networks
2. Explain network models, layers and communication protocols
3. Develop understanding of transmission media, encoding and error control
4. Explain network layer functions and important protocols
5. Provide knowledge of internet concepts and basic network security

Course Outcomes

After completing this course, the student will be able to

CO1: Explain fundamentals of data communication, networks and protocols

CO2: Describe network models, layers and their functions

CO3: Understand transmission media, encoding and error control techniques

CO4: Explain network layer concepts, addressing and protocols

CO5: Understand internet services and basic network security

Week-wise Syllabus Plan

Week	Course Outcomes	Program Outcomes	Topics	Contents
Week 1	CO1	PO1, PO2, PO5, PO12	Introduction to Data Communication	• Data communication concept • Components of data communication • Data flow • Characteristics of data communication
Week 2	CO1	PO1, PO2, PO5, PO12	Computer Networks and Protocols	• Definition of computer networks • Goals and advantages of networks • Types of networks LAN, MAN, WAN • Concept of protocol • Importance of protocols in communication
Week 3	CO2	PO1, PO2, PO3, PO5, PO10, PO12	Network Models and Protocol Architecture	• OSI reference model • Functions of each OSI layer • TCP IP model • Protocol architecture • Comparison of OSI and TCP IP
Week 4	CO3	PO1,PO2, PO3,PO5, PO10,PO12	Physical Layer Concepts	• Role of physical layer • Data and signals • Analog and digital signals • Transmission impairments
Week 5	CO3	PO1, PO2, PO3, PO5, PO10, PO12	Transmission Media	• Guided transmission media twisted pair, coaxial cable, optical fiber • Unguided transmission media, radio waves, microwaves, satellite
Week 6	CO3	PO1, PO2, PO3, PO5, PO10, PO12	Data Encoding and Error Detection	• Digital to digital encoding techniques • Analog to digital conversion • Error detection methods parity check, checksum, CRC
Week 7	CO3	PO1, PO2, PO3, POS, PO10, PO12	Data Link Layer and Protocols	• Functions of data link layer • Framing • Flow control • Error control

				• Data link layer protocols overview
Week 8	CO4	PO1, PO2, PO3, PO4, PO5, PO10, PO12	Network Layer Fundamentals	• Functions of network layer • Logical addressing • lPv4 addressing basics • Subnet concept
Week 9	CO4	PO1, PO2, PO3, PO4, PO5, PO10, PO12	Network Layer Protocols	• Internet Protocol IP • Internet Control Message Protocol ICMP • Address Resolution Protocol ARP • Reverse ARP • Role of routing protocols
Week 10	CO4	POU PO2t PO3, PO4, PO5, PO10, PO12	Transport Layer and Switching Techniques	• Transport layer functions • TCP and UDP • Ports and multiplexing • switching techniques circuit switching packet switching
Week 11	CO5	PO1, PO2, PO3, PO4, PO5, PO9, PO10, PO12	Internet and Application Layer Protocols	• Internet concept • Domain Name System • HTTP and HTTPS • FTP • Email protocols SMTP, POP, IMAP
Week 12	CO5	PO1, PO2, PO3yPO4/ PO5, PO9, PO10, PO12	Wireless Networks and Mobile Communication	• Wireless network concepts • Wi Fi • Bluetooth • Mobile communication basics
Week 13	COS	PO1, PO2, PO3, PO4, Nos, fo9, PO10, PO12	Network Security Basics	• Network security concepts • Types of attacks • Encryption basics • Firewalls and antivirus

Suggested Additional Activities

1. Online Resources and Learning Platforms
   a. GeeksforGeeks — Computer Networks
      https://www.geeksforgeeks.org/computer-network-tutorials
      Clear explanations with diagrams and examples.
   b. TutorialsPoint — Computer Networks
      https://www.tutorialspoint.com/computer_networks
      Very helpful for quick revision and basics.
   c. Cisco Networking Basics
      https://www.cisco.com
      Good for understanding practical networking concepts.
   d. NPTEL Video Lectures
      https://npteI.ac.in
      High quality lectures from Indian professors.
   e. YouTube Channels Gate Smashers Neso Academy Jenny's Lectures
      Good for visual learning and exam preparation.
   f. Software and Tools (Optional Exposure)
      Packet Tracer (Cisco)
      Wireshark (basic packet analysis)

2. Reference Books
   a. Data Communications and Networking — Behrouz A. Forouzan Simple language and very suitable for beginners.
   b. Computer Networks — Andrew S. Tanenbaum
      Good for conceptual understanding of networking principles.
   c. Computer Networks — A Top Down Approach by Kurose and Ross Explains networking using practical examples.
   d. Computer Networking — James F. Kurose
      Useful for understanding protocols and internet concepts.

Web Technology

Course Objectives

The objectives of this course are to

1. Introduce students to internet fundamentals and principles of web design
2. Develop skills in client side web development using HTMLS, CSS3 and JavaScript
3. Provide basic knowledge of responsive design and front end frameworks
4. Introduce server side technologies for dynamic web applications
5. Enable students to work with databases, data formats and web services

Course Outcomes

After completing this course, students will be able to

- CO1: Explain internet fundamentals, web architecture and basic web design principles
- CO2: Design responsive web pages using HTMLS and CSS3
- CO3: Develop interactive client side applications using JavaScript and frameworks
- CO4: Build basic server side web applications using PHP, Servlets and JSP
- CO5: Apply database connectivity and web services concepts in web applications

Week-wise Syllabus Plan

Week	Course Outcomes	Program Outcomes	Topics	Contents
Week 1	CO1	PO1, PO2, PO5, PO6, PO12	Internet fundamentals	• Internet basic terminologies and WWW concepts • HTTP requests and responses • Web browsers and web servers • Web 2.0 concepts
Week 2	CO1	PO1, PO2, PO5, PO6, PO12	Web Design Basics	• Web Design Basics User centric design • Page layout principles • Navigation and site structure • Website planning process
Week 3	CO2	PO1, PO2, PO3, PO5, PO12	HTMLS Structure and Elements	• HTMLS document structure • Head and body elements • Text, images and links • Tables in HTMLS
Week 4	CO2	PO1, PO2, PO3, PO5, PO12	HTMLS Forms and Media	• Form elements and controls • Input validation basics • Audio and video elements • New HTMLS form controls
Week 5	CO2	PO1, PO2, PO3, POS, PO12	CS53 Basics	• CSS syntax • Colors and fonts • Box model • Styling text and images
Week 6	CO2	PO1, PO2, PO3, PO5, PO12	CSS3 Layout and Responsiveness	• Positioning techniques • Flex box layout • Responsive web design • Media queries
Week 7	CO3	PO1, PO2, PO3, PO5, PO9, PO10, PO12	CSS3 Effects and Frameworks	• CSS animations • Transitions • Introduction to Bootstrap • Responsive components in Bootstrap

Week 8	CO3	PO1, PO2, PO3, PO5, PO9, PO10, PO12	JavaScript Basics	• JavaScript syntax • Variables and data types • Operators and functions • Control statements
Week 9	CO3	PO1, PO2, PO3, PO5, PO9, PO10, PO12	JavaScript Advanced Concepts	• DOM manipulation • Event handling • Form validation • DHTML concepts • JSON basics
Week 10	CO4	PO1, PO2, PO3, PO4, PO5, PO9, PO10, PO12	Server Side Web Development	• Introduction to server side scripting • PHP syntax • Variables and control structures • Form handling and validation in PHP
Week 11	CO4	PO1, PO2, PO3, PO4, PO5, PO9, PO10, PO12	Java Based Web Technologies	• Servlet architecture • Servlet lifecycle • Handling GET and POST requests • Sessions and cookies
Week 12	CO4	PO1, PO2, PO3, PO4, POS, PO9, PO10, PO12	JSP and MVC Architecture	• JSP syntax • Scriptlets and expressions • JSP tags • MVC architecture in web application
Week 13	CO5	PO1, PO2, PO3, PO4, PO5, PO9,	Databases, Data Formats and Web Services	• Relational databases and tables • SQL queries using MySQL • JDBC connectivity • XML structure and parsing • Introduction to APIs • Basics of web services and SOA

		PO10, PO12		

Suggested Additional Activities

1. Online Resources and Learning Platforms

   W3Schools: https://www.w3schools.com Covers HTML, CSS and JavaScript with examples and simple explanations.

   - MDN (Mozilla Developer Network): https://developer.mozilla.org Authoritative resource for web standards and detailed explanations.
   - freeCodeCamp: https://www.freecodecamp.org Includes interactive tutorials on HTML, CSS and JavaScript.

   Codecademy (Free Courses): https://www.codecade v com Interactive lessons in web development for beginners.

   YouTube Channels

   - Traversy Media — Practical web development tutorials
   - The Net Ninja — HTML, CSS, JavaScript basics
     - freeCodeCamp.org — Full length beginner friendly courses
   - Online HTML Editor: https://codepen.io / https://|sfiddIe.net For live coding and sharing web pages.
   - Browser Dev Tools: Use Chrome or Firefox inspector to see HTML/CSS changes in real time.

2. Reference Books:

   1. "Web Technology" by Uttam K. Roy
      Good explanation of HTML, CSS and JavaScript basics.
   2. "Web Technology: HTML, JavaScript, PHP, Java, JSP, ASP.NET, XML and AJAX" by Achyut S. Godbole and Atul Kahate
      Covers web fundamentals with clear examples.
   3. "Internet and Web Technologies" by Raj Kamal
      Focuses on both internet basics and web programming.
   4. "HTML, CSS, and JavaScript All in One" by Julie C.
      Meloni Student friendly and project oriented.

Web Technology Programming Lab

Lab Course Objectives

The objectives of this lab are to

1. Provide hands on experience with web design and development tools
2. Enable students to create static and responsive web pages
3. Develop skills in client side scripting using JavaScript
4. Introduce basic server side scripting concepts
5. Familiarize students with databases, data formats and web services

Lab Course Outcomes

After completing the lab, the student will be able to

CO1: Design basic web pages using HTMLS

CO2: Apply CSS3 for styling and responsive layouts

CO3: Develop interactive web pages using JavaScript

CO4: Create basic server side web applications

CO5: Work with databases, XML and basic web services

HTMLS and Web Design Basics (CO1, CO2)

1. Create a simple HTML page to display your personal profile.
2. Design an HTML page using headings, paragraphs, images and hyperlinks.
3. Create an HTML page using ordered, unordered and definition lists.
4. Design an HTML table to display student academic details.
5. Create an HTMLS page using audio and video elements.

HTMLS Forms and Validation (CO2)

6. Design an HTML form with text box, password, radio button, checkbox and submit button.
7. Create a registration form using HTMLS input types with basic validation.
8. Design a login form and apply input validation using HTML attributes.

CSS3 Styling and Responsiveness (CO2)

9. Apply inline, internal and external CSS to an HTML page.
10. Design a web page using CSS box model properties.
11. Create a responsive web page using media queries.
12. Design a webpage layout using CSS

flexbox. CSS Effects and Frameworks (CO3)

13. Apply CSS transitions and animations to a webpage.
14. Design a responsive webpage using Bootstrap components.
15. Create a navigation bar using Bootstrap.

JavaScript Basics and Advanced Concepts (CO3)

16. Write a JavaScript program to perform basic arithmetic operations.
17. Create a JavaScript program to validate form inputs.
18. Write a JavaScript program to manipulate HTML elements using DOM.
19. Develop a Java5cript program to handle mouse and keyboard events.
20. Create a JavaScript program to work with arrays and JSON objects.

Server Side and Database Basics (CO4, CO5)

21. Write a PHP program to process form data and display output.
22. Develop a PHP program to perform simple validation on server side.
23. Write a simple Servlet program to handle GET and POST requests.
24. Create a JSP page to display dynamic content using scriptlets.
25. Design a simple database driven web page using MySQL and JDBC or PHP.

Object Oriented Programming through Java

Course Objectives

The objectives of this course are to

1. Introduce object oriented programming concepts using Java
2. Develop logical and problem solving skills through Java programming
3. Understand core Java features such as classes, inheritance and exception handling
4. Learn multithreading and file handling concepts
5. Provide exposure to basic GUI programming and event handling

Course Outcomes

After completing this course, the student will be able to

CO1: Explain object oriented concepts and Java fundamentals

CO2: Develop Java programs using classes, objects, methods and constructors

CO3: Implement arrays, strings and wrapper classes in Java programs

CO4: Apply inheritance, polymorphism, interfaces and exception handling

CO5: Develop Java applications using multithreading, file handling and basic GUI

Week-wise Syllabus Plan

Week	Course Outcomes	Program Outcomes	Topics	Contents
Week 1	CO1	PO1, PO2, PO5, PO12	Introduction to OOP and Java	• Object oriented paradigm • Comparison of OOP with procedure oriented programming • Features of Java • Java Virtual Machine • JDK and JRE
Week 2	CO1	PO1, PO2, POS, PO12	Java Program Structure and Basics	• Structure of Java program • Data types • Variables • Operators • Control statements if else, switch • Looping statements
Week 3	CO1	PO1, PO2, PO5, PO12	Input Output in Java	• Using Scanner class • Reading input • Displaying output • Simple Java programs
Week 4	CO2	PO1, PO2, PO3, PO5, PO9, PO10, PO12	Classes and Objects	• Defining classes • Creating objects • Instance variables • Local variables
Week 5	CO2	PO1, PO2, PO3, PO5, PO9, PO10, PO12	Methods and Constructors	• Method declaration and calling • Method overloading • Constructors and types of constructors • Constructor overloading • this keyword • super keyword
Week 6	CO2	PO1, PO2, PO3, PO5, PO9, PO10, PO12	Memory Management and Modifiers	• Garbage collection • finalize method • Access modifiers • public, private, protected and default
Week 7	CO3	PO1, PO2, PO3, POS, PO10, PO12	Arrays in Java	• One dimensional arrays • Multi-dimensional arrays • Array based programs

Week 8	CO3	PO1, PO2, PO3, Pos, PO10, PO12	Strings and Wrapper Classes	• String class and its methods • StringBuffer • StringBuilder • Wrapper classes
Week 9	CO4	PO1, PO2, PO3, PO4, Pos, Po9 PO10, PO12	Inheritance and Polymorphism	• Creating subclasses • Types of inheritance • Method overriding • final keyword • Abstract classes and methods • Dynamic method dispatch
Week 10	CO4	PO1, PO2, PO3, PO4, Pos, Po9 PO10, PO12	Encapsulation and Interfaces	• Packages • Access control • Defining and implementing interfaces • Interface variables • Multiple inheritance using interfaces
Week 11	CO4	PO1, PO2, PO3, PO4, PO5, PO9, PO10, PO12	Exception Handling	• Types of exceptions • try catch blocks • finally block • throw and throws • Custom exceptions
Week 12	CO5	PO1, PO2, PO3, PO4, PO5, PO9, PO10, PO12	Multithreading	• Introduction to multithreading • Thread life cycle • Creating threads using Thread class • Runnable interface • Synchronization • Inter thread communication
Week 13	COS	PO1, PO2, PO3, PO4, PO5, PO9, PO10, PO12	File Handling and GUI Programming	• 10 streams • Byte and character streams • File class • Reading and writing files • Introduction to AWT and Swing • Components and containers • Event handling basics

Suggested Additional Activities

1. Online Resources and Learning Platforms
   b. Oracle Java Documentation
      https://docs.oracle.com/javase
      Official and reliable source for Java concepts.
   c. GeeksforGeeks — Java Section
      https://www.geeksforgeeks.org/Adva
      Clear explanations with examples and practice problems.
   d. TutorialsPoint — Java
      https://www.tutorialspoint.com/java
      Simple and structured tutorials for beginners.
   e. W3Schools — Java
      https://www.w3schools.com/java
      Easy to understand Java basics with examples.
   f. YouTube Channels
      Telusko — Core Java explained clearly
      Durga Software Solutions — Strong conceptual teaching
      freeCodeCamp.org — Full Java courses for beginners
   g. Tools and Software Suggested
      JDK (Latest LTS version)
      Eclipse IDE / lnteIIiJ IDEA / NetBeans
      Online Java Compilers for practice

2. Reference Books
   a. "Programming with Java" by E. Balagurusamy
      Very popular book, easy language and good examples.
   b. "Java: The Complete Reference" by Herbert Schildt
      Detailed explanation of core and advanced Java concepts.
   c. "Object Oriented Programming with Java" by R. Nageswara Rao
      Focuses well on OOP concepts using Java.
   d. 'Core Java for Beginners" by Rashmi Kanta
      Das Student friendly book for fundamentals.

Object Oriented Programming through Java Lab

Object Oriented Programming through Java Programming Lab

Lab Course Objectives

The objectives of this lab are to

1. Provide hands on experience in Java programming
2. Strengthen understanding of object oriented concepts
3. Develop problem solving and logical thinking skills
4. Enable students to write, test and debug Java programs
5. Introduce multithreading, file handling and basic GUI programming

Lab Course Outcomes

After completing the lab, the student will be able to

CO1: Write basic Java programs using control statements and input output
CO2: Develop programs using classes, objects, methods and constructors
CO3: Implement arrays, strings and wrapper classes
CO4: Apply inheritance, polymorphism, interfaces and exception handling
CO5: Develop Java programs using multithreading, file handling and GUI

Java Lab Questions

Java Basics and Control Structures (CO1)

1. Write a Java program to display "Welcome to Java Programming".
2. Write a Java program to accept two numbers and perform all arithmetic operations.
3. Write a Java program to check whether a given number is even or odd.
4. Write a Java program to find the largest of three numbers using if else.
5. Write a Java program to generate Fibonacci series using loop.

Classes, Objects, Methods and Constructors (CO2)

6. Write a Java program to define a class and create its object.
7. Write a Java program to demonstrate use of instance and local variables.
8. Write a Java program to demonstrate method overloading.
9. Write a Java program to implement constructor overloading.
10. Write a Java program to demonstrate use of this and super keywords.

Arrays, Strings and Wrapper Classes (CO3)

11. Write a Java program to store and display elements of one dimensional array.
12. Write a Java program to perform matrix addition using two dimensional arrays.
13. Write a Java program to demonstrate commonly used String class methods.
14. Write a Java program to demonstrate StringBuffer and StringBuilder.
15. Write a Java program to demonstrate use of wrapper classes.

Inheritance, Interfaces and Exception Handling (CO4)

16. Write a Java program to demonstrate single inheritance.
17. Write a Java program to demonstrate method overriding and dynamic method dispatch.
18. Write a Java program to demonstrate abstract class and abstract methods.
19. Write a Java program to implement an interface and achieve multiple inheritance.
20. Write a Java program to handle exceptions using try, catch and finally blocks.

Multithreading, File Handling and GUI (COS)

21. Write a Java program to create a thread by extending Thread class.
22. Write a Java program to create a thread using Runnable interface.
23. Write a Java program to demonstrate synchronization in multithreading.
24. Write a Java program to read and write data into a file using File class.
25. Write a Java program to create a simple GUI using AWT or Swing with event handling.

Database Management Systems

Course Objectives

The objectives of this course are to

1. Introduce fundamental concepts of database systems and DBMS architecture
2. Develop understanding of data models and database design techniques
3. Enable students to design ER diagrams and relational schemas
4. Provide strong knowledge of SQL for database creation and manipulation
5. Introduce transaction management, normalization and concurrency concepts

Course Outcomes

After completing this course, the student will be able to

CO1: Explain database concepts, architecture and roles of DBMS users

CO2: Design ER models and relational database schemas

CO3: Apply relational algebra and SQL for database operations

CO4: Normalize database designs and handle advanced SQL features

CO5: Understand transaction management, concurrency control and recovery

Week-wise Syllabus Plan

Week	Course Outcomes	Program Outcomes	Topics	Contents
Week 1	CO1	PO1, PO2, PO5, PO12	Introduction to Databases	• Overview of database systems • Purpose of a database • Characteristics of database approach • Components of a DBMS • Database users • Role of Database Administrator
Week 2	CO1	PO1, PO2, PO5, PO12	Database Architecture	• Three level architecture external schema conceptual schema internal schema • Data independence logical data independence physical data independence
Week 3	CO1	PO1, PO2, POS, PO12	Data Models	• Introduction to data models • ER model • Relational model • Comparison of data models
Week 4	CO2	PO1, PO2, PO3, PO5, PO9, PO10, PO12	Entity Relationship Modelling	• Entities and entity sets • Attributes and types of attributes • Key attributes • Relationships and types of relationships
Week 5	CO2	PO1, PO2, PO3, PO5, PO9, PO10, PO12	Advanced ER Concepts	• Degree of relationships • Cardinality constraints • Weak entity sets • ER diagrams
Week 6	CO3	PO1, PO2, PO3, PO5, PO9, PO10, PO12	Relational Model	• Structure of relational databases • Database schema • Keys, super key, candidate key primary key, foreign key • Integrity constraints entity integrity • referential integrity

Week 7	CO3	PO1, PO2, PO3, PO5, PO10, PO12	Relational Algebra	• Basic operations selection projection union intersection difference • Join operations
Week 8	CO3	PO1, PO2, PO3, PO5, PO10, PO12	SQL Basics	• Introduction to SQL • DDL commands CREATE ALTE R DRO P • DML commands INSERT, UPDATE, DELETE SELECT
Week 9	CO4	PO1, PO2, PO3, PO4, PO5, PO9 PO10, PO12	Advanced SQL	Operators and conditions • Aggregate functions • GROUP BY and ORDER BY • Joins inner join outer join • Subqueries • Views
Wee k 10	CO4	PO1, PO2, PO3, PO4, PO5, PO9 PO10, PO12	PL SQL Concepts	• Introduction to PL SQL • Stored procedures • Functions • Triggers
Wee k 11	CO4	PO1, PO2, PO3, PO4, PO5, PO9, PO10, PO12	Database Design and Normalization	• Database design process • Anomalies in poor design • Functional dependencies • First Normal Form • Second Normal Form
Wee k 12	CO4	PO1, PO2, PO3, PO4, POS, PO9,	Advanced Normalization	• Third Normal Form • Boyce Codd Normal Form • Advantages of normalization

		PO10, PO12		
Week 13	cos	PO1, PO2, PO3/ PO4, PO5, PO9, PO10, PO12	Transaction Management and Concurrency	• Transaction concepts • ACID properties • Serializability • Concurrency control techniques • Locking protocols • Basic recovery systems

Suggested Additional Activities

1. Online Resources and Learning Platforms
   a. GeeksforGeeks — DBMS Section
      https://www.geeksforgeeks.org/dbms
      Simple explanations with examples and practice problems.
   b. TutorialsPoint — DBMS
      https://www.tutorialspoint.com/dbms
      Beginner friendly structured tutorials.
   C. W3Schools — SQL
      https://www.w3schools.com/sql Very good for practicing SQL queries.
   d. Oracle Database Documentation
      https://docs.oracle.com/en/database
      Official reference for SQL and PL SQL concepts.
   e. YouTube Channels
      Gate Smashers — DBMS concepts explained clearly Jenny's Lectures — ER modeling and normalization Neso Academy — DBMS theory and examples
   f. Software Tools Suggested
      MySQL / MariaDB
      Oracle Express Edition (XE)
      phpMyAdmin / SQL Developer

2. Reference Books
    a. “Database System Concepts” by Abraham Silberschatz, Henry F. Korth and S. Sudarshan
    Standard textbook with clear explanation of DBMS fundamentals.
    b. “Fundamentals of Database Systems” by Ramez Elmasri and Shamkant B. Navathe
    Very good for ER modeling, normalization and design.
    c. “Database Management Systems” by Raghu Ramakrishnan and Johannes Gehrke
    Covers database architecture, transactions and concurrency.
    d. “SQL, PL/SQL: The Programming Language of Oracle” by Ivan
    Bayross Useful for SQL and PL SQL practice.

Database Management System Programming Lab

Lab Course Objectives

The objectives of this lab are to

1. Provide hands on experience with database concepts and tools
2. Enable students to design and implement relational databases
3. Develop skills in writing SQL queries for data definition and manipulation
4. Introduce database design techniques using ER modeling and normalization
5. Familiarize students with transaction management concepts

Lab Course Outcomes

After completing this lab, the student will be able to

CO1: Understand database structure, users and DBMS environment

CO2: Design ER diagrams and convert them into relational schemas

CO3: Write SQL queries using DDL and DML commands

CO4: Apply normalization and advanced SQL concepts

CO5: Understand transaction management and concurrency control

Database Basics and Schema Design (CO1, CO2)

1. Create a database for a college management system.
2. Design tables for students, courses and departments with suitable data types.
3. Identify primary keys and foreign keys for the given tables.
4. Draw an ER diagram for a library management system.
5. Convert the ER diagram into relational schema.

DDL Commands (CO3)

6. Create tables using CREATE command with constraints.
7. Modify table structure using ALTER command.
8. Delete a table using DROP command and explain its effect.

DML Commands and Basic SQL Queries (CO3)

9. Insert records into tables using INSERT command.
10. Update records using UPDATE command with conditions.
11. Delete specific records using DELETE command.
12. Retrieve data using SELECT command with WHERE clause.
13. Use operators and conditions in SELECT

queries. Aggregate Functions and Joins (CO4)

14. Write SQL queries using aggregate functions like COUNT, SUM and AVG.
15. Use GROUP BY and ORDER BY clauses in SQL queries.
16. Perform INNER JOIN between two tables.
17. Perform LEFT OUTER JOIN and RIGHT OUTER JOIN.

Subqueries, Views and PL SQL (CO4)

1g. Write SQL queries using subqueries.
19. Create and use views in SQL.
20. Write a simple stored procedure using PL SQL.
21. Create a function in PL SQL.
22. Create a trigger for insert or update operation.

Normalization and Transactions (CO4, CO5)

23. Normalize a given un-normalized table up to Third Normal Form.
24. Identify functional dependencies for a given relation.
25. Demonstrate transaction control using COMMIT and ROLLBACK commands.

SEMESTER-IV

Advance Python Programming

Course Objectives

The objectives of this course are to

1. Strengthen Python programming concepts beyond basics.
2. Introduce advanced data handling and modular programming.
3. Develop skills in file handling, exception handling and OOP using Python.
4. Familiarize students with libraries for data processing and visualization.
5. Prepare students for real world Python based application development.

Course Outcomes

After completing this course, the student will be able to

CO1: Apply advanced Python syntax and built in features effectively.

CO2: Develop programs using functions, modules and packages.

CO3: Implement object oriented programming concepts in Python.

CO4: Handle files, exceptions and external data efficiently.

CO5: Use Python libraries for data processing and basic applications.

Week-wise Syllabus

Week	Course Outcomes	Program Outcomes	Topics	Contents
Week 1	CO1	PO1, PO2, PO5, PO12	Review of Python Basics	• Python programming structure • Variables and data types • Control statements • Writing efficient Python code

Week 2	CO1	PO1, PO2, PO5, PO12	Advanced Data Types	• Lists and advanced list operations • Tuples and sets • Dictionary operations • Comprehensions
Week 3	CO2	PO1, PO2, PO3, PO5, CO10, PO12	Functions in Python	• Defining and calling functions • Arguments and return values • Default and keyword arguments • Lambda functions
Week 4	CO2	PO1, PO2, PO3, PO5, PO10, PO12	Modules and Packages	• Importing modules • Creating user defined modules • Packages and package structure • Using standard Python modules
Week 5	CO3	PO1, PO2, PO3, PO4, PO5, CO10, PO12	Object Oriented Programming Basics	• Classes and objects • Instance and class variables • Methods and constructors
Week 6	CO3	PO1, PO2, PO3, PO4, PO5, CO10, PO12	Advanced OOP Concepts	• Inheritance • Method overriding • Polymorphism • Encapsulation
Week 7	CO3	PO1, PO2, PO3, PO4, PO5, CO10, PO12	Special Methods and Iterators	• Magic methods • Operator overloading • Iterators and generators
Week 8	CO4	PO1, PO2,	Exception Handling	• Types of errors

		PO3, PO4, PO5, CO10, PO12	”	• Try, except and finally blocks • User defined exceptions
Week 9	CO4	PO1, PO2, PO3, PO4, PO5, PO10, PO12	File Handling	• Reading and writing text files • Working with CSV files • File handling using with statement
Week 10	CO4	PO1, PO2, PO3, PO4, PO5, PO10, PO12	Regular Expressions and String Processing	• Regular expression syntax • Pattern matching • String searching and manipulation
Week 11	CO5	PO1, PO2, PO3, PO5, PO10, PO12	Python Libraries for Data Handling	• Introduction to NumPy • Array operations • Introduction to Pandas
Week 12	CO5	PO1, PO2, PO3, PO4, PO5, PO9, PO10, PO12	Data Visualization and Applications	• Introduction to Matplotiib • Plotting graphs • Simple data analysis examples
Week 13	CO5	PO1, PO2, PO3, PO4, PO5, PO9, PO10, PO12	Python Applications and Best Practices	• Python coding standards • Debugging and testing basics • Mini project overview

Suggested Additional Activities

1. **Online Resources and Learning Platforms**
    a. Python Official Documentation
    https://docs.python.org
    Best source for accurate Python information.
    b. GeeksforGeeks — Python https://www.geeksforgeeks.org/python-programming-language Simple explanations and practice problems.
    c. TutorialsPoint — Python
    https://www.tutorialspoint.com/python Good for beginners and revision.
    d. W3Schools — Python
    https://www.w3schools.com/python Easy learning with examples.
    e. NPTEL Video Lectures
    https://nptel.ac.in
    Indian university level Python courses.
    f. YouTube Channels
    CodeWithHarry
    Neso Academy

2. **Reference Books**
    a. Core Python Programming — R. Nageswara Rao
    Very suitable for Indian university students.
    b. Python Programming: A Modern Approach — Vamsi Kurama
    Clear explanation with practical examples.
    c. Learning Python — Mark Lutz
    Detailed coverage of Python concepts.
    d. Programming Python — Mark Lutz
    Useful for advanced and application based learning

Advance Python Programming LAB

Lab Course Objectives

The objectives of this lab are to

1. Strengthen advanced Python programming skills
2. Develop problem solving ability using Python
3. Apply object oriented programming concepts in Python
4. Work with files, exceptions and data handling libraries
5. Build confidence in developing small Python applications **Lab**

Course Outcomes

After completing the lab, the student will be able to

CO1: Write Python programs using advanced data types and functions

CO2: Develop modular programs using modules and packages

CO3: Implement object oriented programming concepts in Python

CO4: Handle files, exceptions and text processing effectively

COS: Use Python libraries for data analysis and simple applications

List of Suggested Lab Programs Advanced

Python Basics (CO1)

1. Write a Python program to demonstrate advanced list operations.
2. Write a Python program using dictionary and set operations.
3. Implement list, dictionary and set comprehensions.
4. Write a Python program using lambda functions.

Functions, Modules and Packages (CO1)

5. Write a Python program using default and keyword arguments.
6. Create a user defined module and import it into another program.
7. Develop a simple package and demonstrate its usage.

Object Oriented Programming (CO2)

8. Write a Python program to create a class and object.
9. Implement constructor and instance variables.
10. Write a Python program to demonstrate inheritance.
11. Write a Python program to demonstrate method overriding.

12. Implement polymorphism using Python c asses.
13. Demonstrate encapsulation using access methods.

Special Features (CO3)

14. Write a Python program using magic methods.
15. Implement operator overloading in Python.
16. Write a Python program using generators and iterators.

Exception Handling and File Handling (CO4)

17. Write a Python program to handle runtime exceptions.
18. Create a user defined exception.
19. Write a Python program to read and write text files.
20. Write a Python program to read and write CSV files.

Data Processing and Visualization (COS)

21. Write a Python program using NumPy arrays and operations.
22. Write a Python program using Pandas for data manipulation.
23. Create basic graphs using Matplotlib.

Mini Application (COS)

24. Develop a menu driven Python application using OOP concepts.
25. Mini project using file handling or data analysis.

Computer Graphics

Course Objectives

The objectives of this course are to

1. Introduce fundamental concepts of computer graphics.
2. Explain graphics hardware and software components.
3. Develop understanding of drawing algorithms and transformations.
4. Familiarize students with 2D and basic 3D graphics concepts.
5. Provide foundation for graphics applications and visualization.

Course Outcomes

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

- CO1: Explain basic concepts, applications and components of computer graphics.
- CO2: Apply line, circle and curve drawing algorithms.
- CO3: Perform 2D transformations and viewing operations.
- CO4: Understand 3D concepts, transformations and projections.
- CO5: Describe advanced graphics concepts and real world applications.

Syllabus Week Wise

Week	Course Outcomes	Program Outcomes	Topic Covered	Detail Contents
Week 1	CO1	1, 2, 5, 12	Introduction to Computer Graphics	Definition of computer graphics, applications of graphics, advantages, classification of graphics, raster and vector graphics.
Week 2	CO1	1, 2, 5, 12	Graphics Hardware and Display Devices	Display systems, raster scan and random scan displays, input devices, graphics software and graphics pipeline

Week 3	CO2	1, 2, 3, 5	Points, Lines and Pixels	Scan conversion, line drawing concepts, DDA line drawing algorithm, Bresenham's line algorithm
Week 4	CO2	1, 2, 3,5	Circle and Curve Drawing	Midpoint circle algorithm, ellipse drawing, introduction to curves, Bezier curves
Week 5	CO3	1, 2, 3,4,5	Two Dimensional Transformations	Translation, scaling, rotation, reflection and shearing, transformation matrices
Week 6	CO3	1, 2, 3, 4, 5	Composite Transformations and Viewing	Composite transformations, window and viewport concepts, 2D viewing pipeline
Week 7	CO3	1, 2, 3, 4, 5	Clipping Algorithms	Line clipping, Cohen Sutherland algorithm, polygon clipping, text clipping
Week 8	CO4	1, 2, 3, 4, 5	Introduction to Three Dimensional Graphics	3D coordinate system, 3D transformations, translation, scaling and rotation in 3D
Week 9	CO4	1, 2, 3, 4, 5	3D Viewing and Projections	Parallel and perspective projections, orthographic projection, projection matrices
Week 10	CO4	1, 2, 3, 4, 5	Visible Surface Detection	Need for visibility detection, depth buffer algorithm, back face detection, painter's algorithm
Week 11	CO5	1, 2, 4, 5, 6,8, 9, 10, 12	Illumination and Shading	Lighting models, ambient, diffuse and specular reflection, flat and Gouraud shading
Week 12	CO5	1, 2, 4, S, 6,8, 9, 10, 12	Graphics Standards and Multimedia	Introduction to graphics standards, OpenGL basics, multimedia concepts, animation basics
Week 13	CO5	1, 2, 4, 5, 6,8, 9, 10, 12	Applications and Recent Trends	Computer graphics applications in games, movies, education, CAD, virtual reality, revision

Online Resources

1. TutorialsPoint — Computer Graphics
   https://www.tutorialspoint.com/computer_graphics Good for quick understanding and revision.
2. GeeksforGeeks — Computer Graphics
   https://www.geeksforgeeks.org/computer-Rraphics Simple explanations and algorithm based programs.
3. NPTEL Video Lectures
   https://nptel.ac.in
   Indian faculty lectures for conceptual clarity.
4. OpenGL Official Documentation
   https://www.opengl.org
   Reference for graphics programming.
5. YouTube Channels
   Gate Smashers
   Neso Academy
   Jenny's Lectures

Reference Books

1. Computer Graphics — Donald Hearn and M. Pauline Baker
   Standard textbook for graphics fundamentals.

2. Computer Graphics — Zhigang Xiang
   Easy explanations with examples.

3. Computer Graphics with OpenGL — Donald Hearn
   Useful for practical implementation.

4. Computer Graphics — Pankaj Sharma
   Suitable for Indian university syllabus.

Computer Graphics Lab Syllabus (Using C Programming)

Lab Course Objectives

The objectives of this lab are to

1. Understand basic computer graphics concepts through programming
2. Implement fundamental graphics algorithms
3. Apply 2D and basic 3D transformations
4. Develop simple graphics based applications
5. Improve logical and visualization skills

Lab Course Outcomes

After completing the lab, the student will be able to

CO1: Use basic graphics functions to draw shapes and patterns

CO2: Implement line, circle and curve drawing algorithms

CO3: Apply 2D transformations and clipping techniques

CO4: Implement basic 3D transformations and projections

CO5: Develop small graphics applications using standard graphics libraries

List of Suggested Lab Experiments (Programs)

Introduction to Graphics

1. Write a program to initialize graphics mode and draw basic shapes.
2. Write a program to draw lines using inbuilt graphics functions.
3. Draw rectangles, circles and ellipses using graphics primitives.

Line and Circle Drawing Algorithms

4. Implement DDA line drawing algorithm.
5. Implement Bresenham's line drawing algorithm.
6. Implement midpoint circle drawing algorithm.
7. Implement ellipse drawing algorithm.

Curves and Filling

8. Draw Bezier curves using control points.
9. Implement boundary fill algorithm.
10. Implement flood fill algorithm.

Two Dimensional Transformations

11. Implement 2D translation of a given objec
12. Implement 2D scaling of a given object.
13. Implement 2D rotation of a given object.
14. Implement reflection and shearing.

Clipping Algorithms

15. Implement Cohen Sutherland line clipping algorithm.
16. Implement polygon clipping algorithm.

Three Dimensional Graphics

17. Write a program for 3D translation.
18. Write a program for 3D scaling.
19. Write a program for 3D rotation.
20. Implement parallel projection.
21. Implement perspective projection.

Mini Applications

22. Design a simple animated scene.
23. Create a simple drawing application.
24. Implement visible surface detection using depth buffer concept.
25. Mini project using graphics concepts.

Cloud Computing

Course Objectives

After completing this course, students will be able to

1. Understand basics of distributed systems and cloud computing
2. Identify real world use of cloud in industries
3. Learn cloud architecture, services, and security
4. Understand how data and services are managed in cloud

Course Outcomes

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

CO1: Explain the basic concepts of distributed systems and cloud computing, including evolution, need, principles, advantages, and industrial applications.
CO2: Describe cloud computing models, including types of clouds and cloud service models used in different application scenarios.
CO3: Explain cloud enabling technologies, cloud architecture, and components involved in cloud infrastructure and networking.
CO4: Understand cloud data management, service management, and security issues involved in cloud environments.
CO5: Explain real world applications of cloud computing in industry and analyze case studies along with recent trends and future scope.

Week Wise Syllabus

Week	Course Outcomes	Program Outcomes	Topics Covered	Detail contents
Week 1	CO1	1, 2, 7	Distributed Systems	Definition, characteristics and examples of distributed systems. Limitations of traditional computing.
Week 2	CO1	1, 2, 7	Evolution to Cloud	Evolution of computing models. Need for shift from distributed systems to

			Computing	cloud computing. Introduction to cloud computing.
Week 3	CO1	1, 2, 7	Cloud Computing Fundamentals	Definition of cloud computing. Real world examples such as Google Drive, AWS and Azure.
Week 4	CO1	1, 2, 7	Importance of Cloud Computing	Why organizations depend on cloud computing. Importance of cloud computing for industries.
Week 5	CO1, C05	1, 2, 5, 6, 7	Principles and Applications	Principles of cloud computing. Advantages and limitations. Applications in engineering and manufacturing.
Week 6	CO3	1, 2, 3, 4	Cloud Enabling Technologies	Virtualization, internet and networking concepts. Role of data centers and web technologies.
Week 7	CO2	1, 2, 3	Types of Cloud	Public cloud, private cloud, hybrid cloud and community cloud with examples.
Week 8	CO2	1, 2, 3	Cloud Service Models	IaaS, PaaS and SaaS. Comparison of service models with use cases.
Week 9	CO3	1, 2, 3, 4	Cloud Architecture	Front end and back end architecture. Components of cloud computing such as clients, data centers and distributed servers.
Week 10	CO3, CO4	1, 2, 3, 4, 5	Cloud Network and Data Management	Role of networking in cloud. Data storage, backup and recovery in cloud environment.
Week 11	CO4	1,2,4,5	Service Management in Cloud	Resource allocation, load balancing and monitoring services in cloud.
Week 12	CO4, CO5	1,2,4, 5, 6, 7	Cyber and Cloud Security	Need for security, security challenges and threats, key components of cyber security, benefits of cyber security, data privacy issues, authentication, authorization, encryption, legal and ethical issues.
Week 13	CO5	1, 5, 6, 7	Case Studies and Future Trends	Cloud usage in IT industry, cloud based ERP systems, cloud computing in education sector and future trends.

Reference Books & Online Resources

1. Cloud Computing: Concepts, Technology & Architecture by Thomas Erl
2. AWS for Beginners by Ryan Kroonenburg
3. Microsoft Azure Fundamentals by Jim Cheshire
4. NPTEL Course on Cloud Computing
5. AWS/Azure Free Tier for hands-on practice

Software Engineering

Course Objectives

The objectives of this course are to

1. Introduce the fundamentals and importance of software engineering
2. Explain software development life cycle models and methodologies
3. Develop understanding of requirements analysis and software design
4. Familiarize students with software testing, quality and maintenance
5. Provide basic knowledge of software project management and ethics

Course Outcomes

After completing this course, the student will be able to

CO1: Explain basic concepts, need and scope of software engineering

CO2: Describe software process models and development methodologies

CO3: Analyze software requirements and apply design techniques

CO4: Understand software testing, quality assurance and maintenance

CO5: Explain basics of software project management and professional ethics

Week-wise Syllabus

Week	Course Outcomes	Program Outcomes	Topics	Contents
Week 1	CO1	PO1, PO2, PO5, PO12	Introduction to Software Engineering	• Definition of software • Characteristics of software • Software crisis • Need and importance of software engineering
Week 2	CO1	PO1, PO2,	Software Engineering Concepts	• Types of software applications

		PO5, PO12		• Software engineering layers • Software process framework • Role and responsibilities of a software engineer
Week 3	CO2	PO1, PO2, PO3, PO5, CO10, PO12	Software Development Life Cycle	• Concept of SDLC • Phases of SDLC • Advantages and limitations of SDLC
Week 4	CO2	PO1, PO2, PO3, PO5, PO10, PO12	Software Process Models	• Waterfall model • Incremental model • Spiral model • Comparison of models
Week 5	CO2	PO1, PO2, PO3, PO5, CO10, PO12	Agile Software Development	• Introduction to Agile • Agile manifesto and principles • Scrum framework • Benefits and limitations of Agile
Week 6	CO3	PO1,PO2, PO3,PO4, PO5, CO10, PO12	Software Requirements Engineering	• Software requirements concepts • Functional and non functional requirements • Requirement elicitation techniques
Week 7	CO3	PO1, PO2, PO3, PO4, PO5, CO10,	Requirement Analysis and SRS	• Requirement analysis process • Software Requirement Specification document

		PO12		• Characteristics of a good SRS
Week 8	CO3	PO1, PO2, PO3, PO4, PO5, CO10, PO12	Software Design Fundamentals	• Design concepts and principles • Architectural design • Modular design • Cohesion and coupling
Week 9	CO3	PO1, PO2, PO3, PO4, PO5, PO10, PO12	Software Design Models	• Data Flow Diagrams • UML diagrams use case diagram class diagram sequence diagram
Week 10	CO4	PO1, PO2, PO3, PO5, PO10, PO12	Software Testing Concepts	• Purpose of software testing • Levels of testing • Black box testing • White box testing
Week 11	CO4	PO1, PO2, PO3, PO5, PO10, PO12	Software Quality and Maintenance	• Software quality attributes • Software quality assurance • Types of software maintenance
Week 12	CO5	PO1, PO2, PO3, PO4, POS, PO9, PO10, PO12	Software Project Management	• Software project planning • Project scheduling techniques • Risk management basics
Week 13	CO5	PO1, PO2, PO3, PO4, PO5, PO9,	Ethics and Modern Trends in Software Engineering	• Professional ethics in software engineering • software

		PO10, PO12		documentation • Component based development • Introduction to DevOps

Suggested Additional Activities

1. **Online Resources and Learning Platforms**
   a. GeeksforGeeks — Software Engineering
   https://www.geeksforgeeks.org/software-engineering

   TutorialsPoint — Software Engineering
   https://www.tutorialspoint.com/software engineering

   b. NPTEL Video Lectures
   https://npteI.ac.in

   c. 4. Coursera / edX (Introductory Courses)
   Introductory courses on software engineering fundamentals.

   d. YouTube Channels
   Gate Smashers
   Neso Academy and Jenny's Lectures

2. **Suggested Reference Books**

   a. Software Engineering — Roger S. Pressman
   One of the best books for understanding fundamentals clearly.
   b. Software Engineering — lan Sommerville
   Covers software processes, design and management.
   c. Software Engineering: A Practitioner's Approach — Pressman & Maxim
   Good balance of theory and practice.
   d. Fundamentals of Software Engineering — Rajib Mall
   Easy to understand for beginners.

PC Assembly and Troubleshooting

Course Objectives

The objectives of this course are to

1. Provide basic understanding of computer hardware components
2. Develop skills in assembling a personal computer system
3. Explain installation and configuration of system software
4. Introduce troubleshooting techniques for hardware and software issues
5. Prepare students for entry level technical support and maintenance roles **Course**

Outcomes (COs)

After successful completion of this course, the student will be able to CO1: Identify and explain components of a personal computer system CO2: Assemble and configure a computer system correctly

CO3: Install operating systems and basic system software

CO4: Diagnose and troubleshoot common hardware and software problems CO5: Apply safety practices, maintenance procedures and documentation skills

Week-wise Syllabus

Week	Course Outcomes	Program Outcomes	Topics	Detail Contents
Week 1	CO1	PO1, PO2, PO5, PO12	Introduction to PC and Computer Hardware	Basic computer organization, input and output devices, CPU, memory, storage devices, ports and connectors
Week 2	CO1	PO1, PO2, PO5, PO12	Motherboard and Processor	Motherboard components, chipsets, BIOS and UEFI, CPU types, sockets, heat sinks and cooling methods
Week 3	CO1	PO1, PO2, PO5, PO12	Memory and Storage Devices	RAM types, cache memory, ROM, hard disk drives, SSD, optical drives, storage interfaces

Week 4	CO1	PO1, PO2, PO5, PO12	Power Supply and Peripheral Devices	SMPS, power ratings, UPS, keyboard, mouse, monitor, printers and scanners
Week 5	CO2	PO1, PO2, PO3, PO5,	PC Assembly Basics	Tools required, safety precautions, step by step PC assembly process, cable management
Week 6	CO2	PO1, PO2, PO3, PO5,	System Configuration and BIOS Setup	BIOS settings, boot sequence, CMOS setup, hardware detection and configuration
Week 7	CO3	PO1, PO2, PO3, PO5,	Operating System Installation	Types of operating systems, installation steps, disk partitioning, driver installation
Week 8	CO3	PO1, PO2, PO3, PO5	Device Drivers and Software Utilities	Role of device drivers, driver installation, system utilities, disk cleanup and optimization
Week 9	CO4	PO1, PO2, PO3, PO4, PO5	Hardware Troubleshooting	Common hardware problems, booting issues, memory and storage faults, troubleshooting tools
Week 10	CO4	PO1, PO2, PO3, PO4, PO5,	Software Troubleshooting	Operating system errors, software conflicts, malware issues, system recovery methods
Week 11	CO5	PO1, PO2, PO3, PO5, PO6, PO8, PO9, PO10, PO12	Preventive Maintenance	Cleaning hardware components, system updates, backup strategies, maintenance schedules
Week 12	CO5	PO1, PO2, PO3, PO4, PO5, PO9,	Safety, Documentation and Electrical safety,	Safety, Documentation and Best Practices Electrical safety, ESD

_____		PO10, PO12	ESD precautions, documentation of issues, maintenance records	precautions, documentation of issues, maintenance records
Week 13	CO5	PO1, PO2, PO3, PO4, PO5, PO9, PO10, PO12	Recent Trends and Revision	Latest PC hardware trends, green computing, energy efficient systems, overall revision

Suggested Additional Activities

1. Online Resources and Learning Platforms
   a. PC Hardware Tutorials
      https://www.tutorialspoint.com/computer fundamentals

   b. HowStuffWorks — Computer Hardware
      https://www.howstuffworks.com

   c. Intel and AMD Official Sites
      Processor architecture and installation guides.

   d. YouTube Channels
      PowerCert Animated Videos
      TechQuickie
      Linus Tech Tips
   e. NPTEL Video Lectures
      https://npteI.ac.in

2. **Suggested Reference Books**
   a. Upgrading and Repairing PCs — Scott Mueller
      Very useful for understanding PC hardware practically.
   b. PC Hardware and Troubleshooting — Mike Meyers
      Simple explanations with practical focus.
   c. Hardware Bible — Winn L. Rosch
      Good reference for PC components.
   d. Troubleshooting and Maintaining PCs — Andrew Bradley
      Useful for beginners.

PC Assembly and Troubleshooting Lab

Lab Course Objectives

The objectives of this lab are to

1. Identify and understand PC hardware components
2. Develop hands on skills in assembling a computer system
3. Perform basic BIOS and system configuration
4. Install operating systems and device drivers
5. Diagnose and troubleshoot common PC hardware and software problems **Lab**

Course Outcomes

After completing this lab, the student will be able to

CO1: Identify PC components and tools used for assembly CO2: Assemble and disassemble a computer system safely CO3: Install and configure operating systems and drivers CO4: Troubleshoot common hardware and software issues

CO5: Apply safety practices and preventive maintenance techniques

List of Suggested Lab Experiments

Familiarization

1. Identify internal and external components of a PC.
2. Identify motherboard slots, ports and connectors.
3. Identify CPU, RAM and storage devices.

PC Assembly and Configuration

4. Demonstrate safe handling of hardware components.
5. Assemble a personal computer step by step.
6. Disassemble and reassemble a PC system.
7. Perform BIOS and CMOS setup.

Software Installation

8. Install a desktop operating system.
9. Create disk partitions and format drives.
10. Install device drivers.
11. Install basic utility software.

Troubleshooting

12. Diagnose booting problems.
13. Troubleshoot memory related issues.

14. Troubleshoot hard disk and storage issues.
15. Identify and fix display and peripheral problems.

Maintenance and Safety

16. Perform preventive maintenance of a PC.
17. Clean internal components safely.
18. Perform system backup and restore.
19. Identify common malware issues and protection methods.
20. Prepare a basic troubleshooting report.