Department of Computer Science and Engineering

Data Structures
15242102

COURSE OBJECTIVES

- To be familiar with the use of data structures as the foundational base for computer solutions to problems.
- To understand various techniques of searching and sorting.
- To understand basic concepts about stacks, queues, lists, trees and graphs.

Unit-I

Introduction to Data Structures, Definition of Data Structure, abstract data types, Classification of Data, Arrays, Various types of Data Structure, Static and Dynamic Memory Allocation.

Link list: Introduction, implementation of linked list, operations, circular link list, doubly linked list, polynomial manipulation using linked list

Unit-II

Stacks: Concepts and implementation of stacks, operations on stack, conversion of infix to postfix notation, evaluation of postfix expression, recursion. Queues: Concepts and implementation, operations on queues, dequeue, priority queues, circular queues and application.

Unit-III

Tree: Types, terminology, binary tree -representations, traversal, conversion of general tree to binary tree, binary search tree, threaded binary tree , red-black tree, and AVL tree.

Graphs: Background, graph theory terminologies, representation of graphs- sequential & linked representation, path matrix.

Unit-IV

Searching & Sorting: Linear search, binary search, bubble sort, selection sort, insertion sort, quick sort, merge sort, radix sort and heap sort, comparison between sorting techniques, hashing and collision resolution techniques.

Unit-V

Advanced Data Structures: Treap (Tree + Heap), Splay Tree, Fenwick Tree / Binary Indexed Tree (BIT), Fibonacci Heap, Perfect Hashing, k-d Tree.

RECOMMENDED BOOKS

- Data Structures, Algorithms and Applications in C++, Sartaj Sahni, 2nd Edition.
- An Introduction to Data Structures with Applications, Jean-Paul Tremblay, Mcgraw hill.
- Data Structures & Algorithms, Aho, Hopcroft & Ullman, original edition, Pearson Publication.

COURSE OUTCOMES

After completion of this course, the students would be able to:

CO1: Understand the fundamental concepts of data structures and implement basic data structures such as arrays and linked lists.

CO2: Apply the principles of stack and queue data structures to solve problems involving expression conversion, evaluation, and recursion etc.

CO3: Analyze tree and graph structures, implement binary search trees, and demonstrate conversion and representation techniques.

CO4: Evaluate and compare the efficiency of various searching and sorting algorithms.

CO5: Understand and implement advanced data structures such as Treaps, Splay Trees, BITs, Fibonacci Heaps, etc.

Course Articulation Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	2	2	2			1	1		1	1	1	1
CO2	3	3	2	2	2			1	2		1	1	2	2
CO3	3	3	2	2	3			1	2		1	1	2	2
CO4	3	3	2	2	2			1	1		1	1	2	3
CO5	3	3	3	2	2			1	2		1	1	3	2

1 - Slightly; 2 - Moderately; 3 – Substantially

Department of Computer Science and Engineering

Database Management System
15242103

COURSE OBJECTIVES

- To understand the fundamental concepts of a database management system.
- To analyze database requirements and determine the entities involved in the system and their relationship to one another.
- To develop the logical design of the database using data modeling concepts & normalization.
- To manipulate a database using SQL commands.

Unit-I

Introduction: DBMS Concepts & Architecture, File processing system, limitation of file processing system, Advantages of Database System, Schemas, Instances, Data Independence, Data dictionary, Functions of DBA, Database languages, Data Models: Hierarchical Data Model, Network Data Model & Relational Data Model, E-R Model, Comparison between Models, Introduction of File organization Techniques.

Unit-II

Relational Data Models: Entities & Attributes, Entity types, Key Attributes, Relationships, Domains, Tuples, types of Attributes, Relations, Characteristics of Relations, Keys, Attributes of Relation, Relational Database, Integrity Constraints.
Relational Algebra: Concept and Relational Algebra operations like Select, Project, Join, Division, Union, etc.

Unit-III

Structure Query Language (SQL): Introduction of SQL, features of SQL, Data Definition & Data Manipulation commands in SQL, SQL operators, Update Statements & Views in SQL, Query & Sub query, Data Retrieval Queries & Data Manipulation Statements examples, etc.

Unit-IV

Normalization: Introduction to Normalization, concepts of anomalies and their types, closure set of dependencies and of attributes, Various Normal Forms: 1NF, 2NF, 3NF, BCNF, Functional Dependency, Decomposition, Dependency Preservation, Loss Less & Lossy Join.

Transaction Processing & Concurrency Control: Transaction Processing Concepts, ACID properties, State Diagram, Types of Transaction, Basic idea of serializability, Concurrency Control, Concurrent operation of Databases, Recovery, Types of Recovery

Unit- V

Basic overview of Distributed Databases System and Relational Database Management System. Introduction to NoSQL, Types of NoSQL Databases, advantages & disadvantages of NoSQL databases, Introduction to Database security.

RECOMMENDED BOOKS

- Abraham Silberschatz, Henry F. Korth, S. Sudarshan, “Database System Concepts”, McGraw-Hill, 6th Edition.
- Raghu Ramakrishnan, Johannes Gehrke, “Database Management System”, McGraw Hill., 3rd Edition.
- Elmasri & Navathe, “Fundamentals of Database System”, Addison-Wesley Publishing, 5th Edition.
- Date C.J, “An Introduction to Database”, Addison-Wesley Pub Co, 8th Edition.
- B.C. Desai, “An introduction to Database Systems”

COURSE OUTCOMES

After completion of this course, the students would be able to:

CO1. Explain the terminology, features, classifications, characteristics, and data models embodied in database systems.
CO2. Describe the concepts related to the data modeling techniques, relational database model and relational algebra.
CO3. Apply the SQL commands over a given relational schema.

CO4. Analyze database schema to apply normalization technique and transaction processing concepts with recovery methods over a real-time database system.
CO5. Analyze the distributed database, NoSQL & Database Security concepts.

CO-PO mapping:

CO	PO													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	1	1	2					1		2	3	2
CO2	3	3	2	2	2					1		2	3	2
CO3	2	3	3	2	3					2		2	3	3
CO4	3	3	2	3	2					2	1	2	3	3
CO5	3	2	2	2	3	2	2	1		2		2	3	3

1 - Slightly; 2 - Moderately; 3 – Substantially

Department of Computer Science and Engineering

Software Engineering
15242104

COURSE OBJECTIVES

- Understand foundational software engineering concepts, including development life cycles, methodologies, and processes.

- Acquire skills in requirements engineering and software design to develop scalable, maintainable, and robust software systems.

- Apply software testing methodologies to ensure the quality, reliability, and effectiveness of software solutions.

Unit – I:

Definition of software engineering, Layered technology, Software characteristics and components, Software development life cycle (SDLC), Waterfall model, Iterative Waterfall model, Prototyping model, Spiral model, RAD model, Selection criteria of models (characteristics of requirements, status of development team, user participation, project type, and associated risks).

Unit – II:

Definition and activities of requirement engineering, Types of requirements (functional and non-functional), User and system requirements, Requirement elicitation methods, Requirement analysis methods, Requirement documentation (SRS), Requirement validation, Requirement management.

Unit – III:

Design fundamentals and principles, Effective modular design, Design representations, Architectural design, Procedural design, Data-directed design, Real-time design, Object-oriented design, Coupling and cohesion, Software metrics, Project management basics (people, product, process, project), Software project estimation (function point estimation, LOC-based estimation, empirical estimation, COCOMO model), Project scheduling techniques.

Unit – IV:

Software testing definitions and life cycle (STLC), Test case design, Strategic approach to software testing (verification & validation), Testing criteria, Unit testing, Integration testing, Validation testing, System testing, Black box and white box testing techniques, Acceptance testing, Introduction to security aspects, Ethical AI in software engineering, Sustainable software lifecycle management.

Unit – V:

Introduction to Agile methodologies, Agile Manifesto and principles, Agile vs traditional methodologies, Key Agile frameworks (Scrum, Kanban, Extreme Programming), Roles in Agile (Scrum Master, Product Owner, Development Team), Agile ceremonies (Sprint Planning, Daily Standups, Sprint Reviews, Retrospectives), Agile metrics (Velocity, Burn Down Chart), Agile tools (JIRA, Trello), Benefits and challenges of Agile adoption.

RECOMMENDED BOOKS

- Software Engineering, Sommerville, Pearson.
- Software Engineering: A Practitioner's Approach, Roger S. Pressman, McGraw Hill.
- Software Engineering, K.K. Agrawal & Yogesh Singh, New Age Publication.
- Software Engineering, Rajib Mall, PHI.

COURSE OUTCOMES

After completion of the course students would be able to:

CO1. **Analyze** software engineering concepts and **evaluate** SDLC models for different projects.
CO2. **Evaluate** requirement engineering techniques and **create** software requirement specifications (SRS).
CO3. **Analyze** design principles and **create** software architectures and project management estimates
CO4. **Evaluate** testing strategies and **create** test plans and cases for various testing types.
CO5. **Evaluate** Agile methodologies and **create** an Agile framework using tools and ceremonies.

Course Articulation Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	3	2	2	1	–	1	1	1	1	1	2	2
CO2	2	3	3	3	2	1	1	–	1	1	1	1	3	3
CO3	2	2	3	3	2	1	1	1	1	1	–	2	3	2
CO4	3	3	2	2	3	1	1	1	–	1	1	2	2	3
CO5	3	2	2	2	3	1	1	1	1	–	1	1	3	2

1 - Slightly; 2 - Moderately; 3 – Substantially

Department of Computer Science and Engineering

Computer Networks
15242105

COURSE OBJECTIVES

- Build an understanding of the fundamental concepts of computer networking.
- Familiarize the student with the basic taxonomy and terminology of the computer networking area.
- Introduce the student to advanced networking concepts, preparing the student for entry Advanced courses in computer networking.

Unit-I

Introduction: Computer Network, Types- LAN, MAN & WAN,Data transmission modes- Serial & Parallel, Simplex, Half duplex & full duplex, Synchronous & Asynchronous transmission, Transmission medium- Guided & Unguided, Cables- Twisted pair, Coaxial cable & Optical fiber, Networking devices- Repeaters, Hub, Switch, Bridge, Router, Gateway, Modem, Proxy Server, Wireless router, & Wireless Access Point (WAPs). Performance Criteria- Bandwidth, Throughput, Latency (Delay), Propagation Time, Transmission time & Queuing Time, Network Standardization- OSI Reference Model & TCP/IP Reference Mode.

Unit-II

Physical Layer and Data Link Layer: Network topologies- Bus, Ring, Star Topology & Mesh, Switching- Circuit switching, Message switching & Packet switching, Multiplexing; FDM – Frequency division multiplexing, WDM – Wavelength division multiplexing & TDM – Time division multiplexing, Wireless transmission- Electromagnetic spectrum, Radio transmission & Microwave transmission.

Introduction, Design issues, Services, Framing, Error control, Flow control, ARQ Strategies, Error Detection and correction, Parity bits, Cyclic Redundant Code (CRC), Hamming codes, MAC Sub Layer- The channel allocation problem, Pure ALOHA ,Slotted ALOHA, CSMA ,CSMA/CD,CSMA/CA,IEEE 802.3 frame format.

Unit-III

Network Layer& Transport Layer: Introduction, Design issues, Services, Routing- Distance vector routing, Hierarchical routing , Link state routing, Shortest path algorithm- Dijkstra's Algorithm & Floyd–Warshall's Algorithm, Flooding, Congestion Control- Open

Loop & Closed Loop Congestion Control, Leaky Bucket & Token bucket Algorithm. Connection Oriented & Connectionless Service, Port addressing basics

Unit-IV

Presentation, Session& Application Layer: Introduction, Design issues, Presentation layer- Translation, Encryption & Compression. Session Layer – Dialog Control, Synchronization. Application Layer- Remote login, File transfer & Electronic mail.

Unit-V

Software-Defined Networking (SDN), Network Function Virtualization (NFV), Green Networking, Quantum Networking, Programmable Networks - Introduction to P4, SmartNICS and P4 switches, Data

Center Networking (DCN) – Introduction, DCN - Deep Dive (Network topologies, Container Network Interfaces)

RECOMMENDED BOOKS

- Behrouz A. Forouzan "Data Communication and Networking", McGraw – Hill Publications.
- Andrew Tanenbaum – Computer Networks, PHI
- Peterson and Davie, "Computer Networks, A systems Approach", 5th ed., Elsevier, 2011.
- Ying-Dar Liu, Ren-Hwang, Fred Baker, "Computer Networks: An open Source Approach", McGraw – Hill, 2001.

COURSE OUTCOMES

After completion of the course students would be able to:

CO1. Explain basic concepts of computer networks, types, transmission modes, transmission media, and performance metrics.

CO2. Illustrate the functions of networking devices and compare the OSI and TCP/IP models.

CO3. Apply data link layer techniques for framing, error control, flow control, and MAC protocols.

CO4. Analyze routing algorithms, addressing techniques, and congestion control mechanisms.

CO5. Demonstrate the use of upper-layer protocols and emerging networking technologies

---

Course Articulation Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	1								2		3	2
CO2	3	3	2	1						2		1	1	1
CO3	3	3	3	2	2	2				2	2	1	2	1
CO4	3	3	3	3	3	3	1			2	2	1	2	1
CO5	2	2	2	1	1	2	2	1	2	2	2	2	1	1

1 - Slightly; 2 - Moderately; 3 – Substantially

Department of Computer Science and Engineering

Data Structures Lab
15242106

List of Experiments

Write C/C++ Programs to illustrate the concept of the following:

1. Implementation of Array and linked list.
2. Implementation of Sorting Algorithms-Non-Recursive and Recursive.
3. Implementation of Searching Algorithms-Linear and Binary Search.
4. Implementation of Stack using Array.
5. Implementation of Queue using Array.
6. Implementation of Circular Queue using Array.
7. Implementation of Stack using Linked List.
8. Implementation of Queue using Linked List.
9. Implementation of Circular Queue using Linked List.
10. Implementation of Tree Structures, Binary Tree, Tree Traversal, Binary Search Tree, Insertion and Deletion in BST.
11. Graph Implementation, BFS, DFS, Minimum cost spanning tree, shortest path algorithm.

COURSE OUTCOMES

After completion of the course students would be able to:

CO1: Apply data structures such as arrays, linked lists, stacks, and queues for efficient data storage and manipulation.

CO2: Develop and analyze sorting and searching algorithms using both recursive and non-recursive approaches.

CO3: Implement and apply tree and graph data structures to solve real-world problems involving hierarchical and networked data.

Course Articulation Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	2	3	2			1	1		1	1	2	3
CO2	3	3	2	2	2			1	2		1	1	3	2
CO3	3	3	2	2	3			1	1		1	1	2	2

1 - Slightly; 2 - Moderately; 3 – Substantially

Department of Computer Science and Engineering

Database Management System Lab
15242107

List of Experiments

1. Introduction to Structure Query Language (SQL)
   - **a)** Overview of SQL
   - **b)** Various Data Types in SQL
   - **c)** Various Commands in SQL
2. Implementation of DDL commands of SQL with suitable examples
   - **a)** Create table
   - **b)** Alter table
   - **c)** Drop table
3. Implementation of DML commands of SQL with suitable examples
   - **a)** Insert
   - **b)** Update
   - **c)** Delete
4. Study and implementation of different types of constraints.
5. Implementation of different types of functions with suitable examples
   - **a)** Number function
   - **b)** Aggregate function
   - **c)** Character function
   - **d)** Conversion function
6. Implementation of different types of operators in SQL
   - **a)** Arithmetic operators
   - **b)** Logical operators
   - **c)** Comparison operators
   - **d)** Set operation
7. Implementation of different types of joins
   - **a)** Inner join
   - **b)** Outer join
   - **c)** Natural join
8. Study and implementation of
   - **a)** Group by and having clause
   - **b)** Order by clause
   - **c)** Indexing
9. Study and implementation of
   - **a)** Sub-queries
   - **b)** Views
10. Study and implementation of Database Backup and Recovery commands.
11. To implement triggers that automatically execute when a specified event occurs.
12. To understand ACID properties and manage transactions using COMMIT, ROLLBACK, and SAVEPOINT.
13. CRUD Operations and Aggregation in MongoDB – A NoSQL Document Database

COURSE OUTCOMES LAB:

After completion of this course lab, the students would be able to:

CO1. Apply basic and advanced SQL commands (DDL, DML, DCL, TCL) to create, modify, query, and manipulate relational database structures using appropriate constraints, operators, and functions.

CO2. Demonstrate the ability to retrieve and manage data using joins, sub-queries, indexing, views, and clauses such as GROUP BY, HAVING, and ORDER BY to support efficient database operations and reporting.

CO3. Implement data integrity and recovery techniques using backup and recovery commands, and explain their significance in ensuring database security and resilience.

CO-PO mapping:

CO	PO													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	2	1	3				1	2	1	2	3	2
CO2	3	3	2	2	3				1	2	1	2	3	3
CO3	2	2	2	2	2	2	1	1	1	2	1	3	3	3

1 - Slightly; 2 - Moderately; 3 – Substantially

Department of Computer Science and Engineering

Problem Solving through Python Programming
15242108

Course Objectives:

- To apply various Python datatypes and Control Structure
- To Implement Classes and objects in Python
- To develop Python GUI.

Unit I

Introduction to Python: Setting up the Python environment (Anaconda, Jupyter Notebook), Basic syntax usage: variables, data types, and operators, First Python program, **Control Structures:** Conditional statements: if, elif, else, Looping constructs: for and while loops Nested control structures

Unit II

List/ Set/ Tuple operations: List, List Operations (Access, Slice, Append, Delete, Unpack, Loop etc), Tuple, Tuple operations (Access, Append, Delete, Unpack, Loop, etc.), Set, Set Operations (Access, Append, Delete, Method, Loop, etc.), Dictionary (Access, Append, Delete, Methods, Loop, etc.), Array (Access, Append, Delete, Methods, Loop, etc.) **Strings:** Reverse, Palindrome, Character count, Replacing Character

Unit III

Matrix and Array: Define matrix and print, Arithmetic operation between Matrix,

Functions and Modules implementation: Defining and calling functions, Function parameters and return values, Scope and lifetime of variables, Importing and using modules, **In-Built Functions, Recursion,** Lambda function

Unit IV

Classes, and Objects: Create class and object, Self-parameter, Attribute and methods, Implement Inheritance and polymorphism.

File Handling: Read and write to files, Working with CSV and JSON file, Implement try-except blocks, Debug a piece of code

Unit V

GUI: Work with Canvas, draw geometric shapes, Fill colour, Creating Simple GUI, GUI packages, Tkinter, Buttons, Labels, Entry Fields, Dialogs; Widget Attributes - Sizes, Fonts, Colours, Layouts, Nested Frames, Widget window – Bg, Bd, Cursor, font, Fg, Command, Minimal Application

Reference Books

- "Python for Data Science For Dummies" by John Paul Mueller and Luca Massaron
- "Python Machine Learning" by Sebastian Raschka and Vahid Mirjalili
- "Python Web Scraping Cookbook" by Michael Heydt

- "Python GUI Programming Cookbook" by Burkhard A.
- "Python for Finance" by Yves Hilpisch.

Course Outcome

CO1: Implement Python built-in functions and control statements.

CO2: Implement Python user-defined functions and classes.

CO3: Create Python GUI.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	3	1	3		1	1	1	1	1	3	3	2
CO2	3	3	3	2	3		1	1	1	1	1	3	3	2
CO3	2	2	3	3	3		1	1	2	3	3	3	3	2

Department of Computer Science and Engineering

Problem Solving through Python Programming

15242108

List of Experiments

1. Python Program to
    a. Generate a Random Number
    b. Convert Kilometres to Miles
    c. Check if a Number is Positive, Negative or 0
    d. Print the Fibonacci sequence
    e. Find ASCII Value of Character
    f. Shuffle Deck of Cards
    g. Display Calendar
2. Python Program to
    a. Display the Fibonacci Sequence Using Recursion
    b. Find the Sum of Natural Numbers Using Recursion
    c. Find the Factorial of Number Using Recursion
    d. Convert Decimal to Binary Using Recursion
3. Python Program to Add Two Matrices, Transpose a Matrix, Multiply Two Matrices
4. Python Program to
    a. Check Whether a String is Palindrome or Not
    b. Remove Punctuations from a String
    c. Sort Words in Alphabetic Order
5. Python Program to Illustrate Different Set, Tuple, and List operations.
6. Python Program to Iterate Over Dictionaries Using for Loop
7. Python Program to Catch Multiple Exceptions in One Line
8. Python Program to Copy a File
9. Python Program to Get Line Count of a File
10. Python Program to Find All Files with .txt Extension Present Inside a Directory
11. Python Program to Return Multiple Values from a Function
12. Write a Python program to create a person class. Include attributes like name, country, and date of birth. Implement a method to determine the person's age
13. Write a Python program to create a class representing a bank. Include methods for managing customer accounts and transactions.
14. Write a Python program to create a class representing a shopping cart. Include methods for adding and removing items and calculating the total price.
15. Create Python GUI using Tkinter
    a. Displaying Text and Images with Label Widgets
    b. Displaying Clickable Buttons with Button Widgets
    c. Getting User Input with Entry Widgets
    d. Getting Multiline User Input with Text Widgets
    e. Assigning Widgets to Frames with Frame Widgets
    f. Adjusting Frame Appearance with Relief

Macro Project-I
15242110

COURSE OBJECTIVES:

- To develop fundamental programming skills through hands-on project implementation
- To foster computational thinking and problem-solving abilities in first-year students
- To introduce students to software development methodologies and collaborative coding practices

---

UNIT 1: INTRODUCTION TO PROGRAMMING FUNDAMENTALS

Experiments:

1. Setting up the development environment (IDE installation and configuration)
2. Basic syntax and data types
3. Control structures (if-else, loops, switch-case)

Projects:

1. Calculator application with basic arithmetic operations
2. Temperature converter (Celsius to Fahrenheit and vice versa)
3. Simple interest and compound interest calculator
4. Number guessing game with user feedback.
5. Student Record Management System
6. Bank Account Simulation
7. Tic-Tac-Toe Game (Console-Based)
8. Prime Number and Factorial Calculator

UNIT 2: DATA STRUCTURES BASICS

Experiments:

1. Arrays and array manipulation
2. Implementing strings and string operations
3. Introduction to linked lists
4. Working with stacks and queues

Projects:

1. Student record management system using arrays
2. Text analyser (word count, character frequency, etc.)
3. To-do list application with basic CRUD operations
4. Implementation of a simple stack-based calculator
5. Library Book Issue System
6. Sorting Algorithms Visualizer

7. Maze Solver Using Stack or Queue

UNIT 3: FILE HANDLING AND DATA PERSISTENCE

Experiments:

1. Reading and writing text files
2. Working with CSV data
3. Exception handling in file operations

Projects:

1. Contact management system with file-based storage
2. Basic notepad application
3. CSV data analyser and visualizer
4. File encryption/decryption tool
5. Student Report Card Generator
6. Railway Reservation System with File Persistence
7. Employee Payroll Management System
8. Mini Quiz System with Result Storage

UNIT 4: GRAPHICAL USER INTERFACE DEVELOPMENT

Experiments:

1. Introduction to GUI libraries
2. Working with buttons, labels, and text fields
3. Event handling and user interaction
4. Layout management

Projects:

1. Stopwatch/timer application with GUI
2. Simple drawing application
3. Quiz application with score tracking
4. Basic image viewer with navigation features
5. Simple Paint Application
6. GUI-Based Calendar and Event Reminder
7. File Explorer
8. Text Editor (Mini Notepad)

UNIT 5: MACRO PROJECT INTEGRATION

Experiments:

1. Project planning and requirements gathering
2. Version control using Git
3. Documentation and code quality
4. Testing and debugging

Projects:

1. Weather forecast application using API
2. E-commerce product catalog
3. Library management system
4. Simple chat application
5. Packet Sniffer using Python and Scapy
6. Network Traffic Monitoring System
7. Custom DHCP Server Simulation
8. Email Server Simulation (SMTP/POP3)
9. Bank Management System
10. Billing System for Retail Shop
11. Cab Booking System

RECOMMENDED BOOKS:

- **"Starting Out with Python" by Tony Gaddis**
- **"Clean Code: A Handbook of Agile Software Craftsmanship" by Robert C. Martin**
- **"Python Crash Course" by Eric Matthes**

COURSE OUTCOMES:

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

CO1: Design and implement small-scale programming projects using appropriate programming concepts and data structures

CO2: Apply problem-solving techniques to develop algorithmic solutions for real-world problems

CO3: Collaborate in teams to design, implement, test, and document software applications while adhering to project timelines and quality standards

---

Course Articulation Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	3	1	1	3	-	-	2	1	2	2	3	2
CO2	2	3	2	2	2	1	1		1	1	1	2	3	3
CO3	1	1	2	-	2	-	-	1	3	2	3	1	2	1

1 - Slightly; 2 - Moderately; 3 – Substantially

CYBER SECURITY

15242112

(Mandatory Audit Course)

COURSE OBJECTIVES

- To introduce the basic concepts of cybersecurity.
- To make students aware of various types of cyber threats, vulnerabilities, security policies and cybersecurity tools.
- To build basic skills for protecting information systems.

Unit I

Introduction to Cyber Security: Overview of Cyber Security, Goals of Cyber Security (Confidentiality, Integrity, Availability), Types of cyber attacks: Phishing, Malware, Ransomware, Social Engineering, Malicious Softwares. Hacker and its types. Real-world incidents and their impact, Cyber Ethics and Legal Aspects.

Unit II

Basics of Networking: Internetworking devices, Topologies OSI and TCP/IP models, IP address, DNS, TCP, IP, HTTP, HTTPS, Web Browser, Web Server.

Unit III

Security Mechanisms: Firewalls, Anti-virus, Intrusion Detection Systems (IDS), Intrusion Prevention Systems (IPS), Encryption and Decryption: Symmetric and Asymmetric, Cryptanalysis, Digital Signature, Authentication: Passwords, Biometrics, Multi-Factor Authentication.

Unit IV

System and Application Security: Operating System security basics. Securing mobile devices and apps. Web application vulnerabilities: SQL Injection, XSS, CSRF. Secure coding practices.

Cybercrime, Forensics, and Incident Response: Types of cybercrimes: Identity Theft, Financial Fraud, Cyberbullying. Basics of digital forensics. Cyber law and IT Act (India) overview. Incident response lifecycle and reporting.

Unit V

Cyber Hygiene and Best Practices: Cyber hygiene: Safe browsing, regular updates, backups. Strong password

creation and management. Social media safety. Roles of individuals and organizations in ensuring security.

RECOMMENDED BOOKS

1. "Cybersecurity for Beginners" by Raef Meeuwisse – Wiley
2. "Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives" by Nina Godbole and Sunit Belapure – Wiley India
3. "Computer Security: Principles and Practice" by William Stallings and Lawrie Brown – Pearson
4. "Introduction to Cyber Security" by Chwan-Hwa (John) Wu and J. David Irwin – CRC Press
5. "Cybersecurity Essentials" by Charles J. Brooks, Christopher Grow, Philip Craig, Donald Short – Wiley

COURSE OUTCOMES

After completion of the course students will be able to:

CO1 Describe fundamental concepts of cyber security and identify common cyber threats and legal implications.

CO2 Explain basic networking concepts.

CO3 Demonstrate common security mechanisms used to protect digital data.

CO4 Analyze cybercrime scenarios and vulnerabilities in systems, and outline procedures for incident response and digital forensics.

CO5 Formulate cyber hygiene strategies and practice safe online behavior to minimize cyber risks.

CO-PO Mapping Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	2		2	1	2		3		2		2
CO2	3	2			2							1
CO3	3	2	2		3	1	1					2
CO4	3	3		3	3	2	1	2		1		2
CO5	2	1		2	2	3	2	3	2	2	1	3

1 - Slightly; 2 - Moderately; 3 – Substantially