GAYATRI VIDYA PARISHAD COLLEGE OF ENGINEERING FOR WOMEN
(Autonomous)
(Approved by AICTE, New Delhi and Permanently Affiliated to Andhra University, Visakhapatnam)
Madhurawada :: Visakhapatnam – 530 048

COURSE STRUCTURE

(Applicable for the Academic Year 2025-26 onwards)
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

B.Tech. CSE-Cybersecurity
SCHEME AND SYLLABI
(With effect from 2025-26 admitted batch)

II Year - I Semester								
Course Code	**Category**	**Course Title**	**Hours per Week**		**Internal Marks**	**External Marks**	**Total Marks**	**Credits**
			L	**P**				
24BM11RC07	BS	Discrete Mathematical Structures for Cybersecurity	3	0	30	70	100	3
24CT11RC11	PC	Computer Organization	3	0	30	70	100	3
24CT11RC12	PC	Database Management Systems	3	0	30	70	100	3
24CC11RC01	PC	Data Structures and Algorithms	3	0	30	70	100	3
24CT11RC13	PC	Operating Systems	3	0	30	70	100	3
24CT11RC16	PC	Database Management Systems Lab	0	3	50	50	100	1.5
24CC11RC02	PC	Data Structures and Algorithms Lab	0	3	50	50	100	1.5
24CT11RC17	PC	Operating Systems Lab	0	3	50	50	100	1.5
24CC11SC01	SC	Web Application Security	1	2	50	50	100	2
24BC11MC01	MC	Environmental Science	2	0	-	-	-	0
							Total Credits	**21.5**

DISCRETE MATHEMATICAL STRUCTURES FOR CYBERSECURITY

II Year B. Tech. I semester
[CSE(Cybersecurity)]

Course Code: 24BM11RC07

L	T	P	C
3	0	0	3

Course Outcomes: At the end of the Course, the student shall be able to

CO1: **Analyze** logical propositions and predicates using truth tables, inference rules, and quantifiers to establish the validity of mathematical arguments. (L4)

CO2: **Apply** set-theoretic concepts, relations, functions, and partial orderings to model and solve discrete computational problems. (L3)

CO3: **Classify** and examine algebraic structures such as semigroups, monoids, groups, rings, and fields based on their defining properties. (L4)

CO4: **Apply** coding-theoretic concepts including parity checks, Hamming distance, and group codes for error detection and correction in digital communication systems. (L3)

CO5: **Analyze** graphs and trees using graph traversal, connectivity, graph isomorphism, and minimum spanning tree algorithms to solve network-related problems. (L4)

UNIT-I **10 Lectures**

MATHEMATICAL LOGIC: Propositional Calculus: Statements and Notations, Connectives, Well Formed Formulas, Truth Tables, Tautologies, Equivalence of Formulas, Duality Law, Tautological Implications, Normal Forms, Theory of Inference for Statement Calculus, Consistency of Premises, Indirect Method of Proof, Predicate Calculus: Predicates, Predicative Logic, Statement Functions, Variables and Quantifiers, Free and Bound Variables, Inference Theory for Predicate Calculus.

(Sections 1.1 to1.6 of Chapter 1 of Text book 1)

UNIT-II **10 Lectures**

SET THEORY AND RELATIONS: Basic concepts of set theory, some operations on Sets, Principle of Inclusion-Exclusion, Relations: Properties, Operations, Partition and Covering, Transitive Closure, Equivalence, Compatibility and Partial Ordering, Hasse Diagrams, Functions: Bijective, Composition, Inverse, Recursive Functions.

(Sections 2.1.1 to 2.1.4, 2.1.9, 2.3.1 to 2.3.9, 2.4.1 to 2.4.3, 2.6.1 of Text book 1)

UNIT-III **10 Lectures**

ALGEBRIAC STRUCTURES: Introduction, Algebraic structures, Semigroups, Monoids, Groups, Subgroups, Cosets, Lagrange's theorem, Cyclic Groups, Algebraic structures with two binary operations: Rings, Field.

(Sections 3.1.1, 3.1.2, 3.2.1,3.2.2, 3.5.1 to 3.5.3, 3.5.5 of Text book 1)

UNIT-IV **10 Lectures**

CODING THEORY: Coding of Binary Information and Error Detection: Message, word, encoding function, Parity check code, Hamming distance, Properties of distance function, Group Codes, Minimum distance, Boolean Matrix, Parity check matrix, Decoding and Error Correction.

(Sections 11.1, 11.2 of Text book 2)

UNIT-V **10 Lectures**

GRAPH THEORY: Graphs: Graph Terminology and Special Types of Graphs, Representing Graphs and Graph Isomorphism, Connectivity, Euler and Hamilton Paths, Shortest-Path Problems, Planar Graphs, Graph Coloring.

(Sections 10.1 to 10.8 of Text book 3)

Trees: Introduction to Trees, Applications of Trees, Tree Traversal, Spanning Trees, Minimum Spanning Trees, Breadth-First Search, Depth-First Search, Prim's and Kruskal's algorithms.

(Sections 11.1 to 11.5 of Text book 3)

Text Books:

1. Discrete Mathematical Structures with Applications to Computer Science, J. P. Tremblay and P. Manohar, Tata McGraw Hill.
2. Discrete Mathematical Structures, Bernand Kolman, Robert C. Busby and Sharon Cutler Ross, 6th Edition, PHI.
3. Discrete Mathematics and its Applications with Combinatorics and Graph Theory, K. H. Rosen, 7th Edition, Tata McGraw Hill.

Reference Books:

1. Discrete Mathematics for Computer Scientists and Mathematicians, J. L. Mott, A. Kandel and T. P. Baker, 2nd Edition, Prentice Hall of India
2. Discrete Mathematics, S. K. Chakraborthy and B. K. Sarkar, Oxford, 2011.
3. Theory and Problems of Discrete Mathematics, Schaum's Outline Series, Seymour Lipschutz and Marc Lars Lipson, 3rd Edition, McGraw Hill.
4. Elements of Discrete Mathematics-A Computer Oriented Approach, C. L. Liu and D. P. Mohapatra, 3rd Edition, Tata McGraw Hill.
5. Discrete and Combinatorial Mathematics by Ralph. G. Grimaldi, Pearson Education, New Delhi.

COMPUTER ORGANIZATION

II Year B. Tech. I semester

[Common to CSE, CSE (AI&ML), IT, CSE(Cybersecurity)]

Course Code: 24CT11RC11

L	T	P	C
3	0	0	3

Course Outcomes: At the end of the Course, the student shall be able to

CO1: **Demonstrate** the design of major components of a basic computer such as register, transfer Micro-operations and arithmetic logic shift unit. (L2)

CO2: **Classify** various instruction codes, timing & control unit and understand the design of CPU, Micro-programmed control Unit. (L2)

CO3: **Make Use of** register organization, instruction formats, and addressing modes to understand the working of central processing units (CPUs) and 8086 microprocessors. (L3)

CO4: **Interpret** different ways of communication with I/O devices and parallel processors. (L2)

CO5: **Outline** the performance of various memory modules and to comprehend computer arithmetic algorithms. (L2)

UNIT-I **08 Lectures**

Register Transfer and Micro operations: Register Transfer Language, Register Transfer, Bus and Memory Transfers, Arithmetic Micro operations, Logic Micro-operations, Shift Micro operations, Arithmetic Logic Shift Unit.

UNIT-II **10 Lectures**

Basic Computer Organization and Design: Instruction Codes, Computer Registers, Computer Instructions, Timing and Control, Instruction Cycle, Memory-Reference Instructions, Input-Output and Interrupt, Complete Computer Description, Design of Basic Computer, Design of Accumulator Logic.

Micro Programmed Control: Control Memory, Address Sequencing, Micro program Example, Design of Control Unit.

UNIT-III **12 Lectures**

Central Processing Unit: Introduction, General Register Organization, Stack Organization, Instruction Formats, Addressing Modes.

Data Transfer and Manipulation, Program Control, Reduced Instruction Set Computer (RISC).

8086 Microprocessor: Family Overview, Internal Architecture.

UNIT-IV **10 Lectures**

Pipeline and Vector Processing: Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline.

Input/Output Organization: Peripheral Devices, I/O interface, Asynchronous data transfer, Modes of transfer, priority Interrupt, Direct memory access.

UNIT-V **10 Lectures**

Computer Arithmetic: Addition and Subtraction with Signed-Magnitude Data, Booth Multiplication Algorithm.

Memory Organization: Memory Hierarchy, Main memory, Associative Memory, Cache Memory.

Text Books:

1. Computer System Architecture, M. Morris Mano, Prentice Hall of India Pvt. Ltd., Third Edition, Sept.2008.
2. Microprocessors and Interfacing, D. V. Hall, TMGH, 2nd Edition 2006.

Reference Books:

1. Computer Architecture and Organization, William Stallings, PHI Pvt. Ltd., Eastern Economy Edition, Sixth Edition, 2003.
2. Computer Organization and Architecture, Linda Null, Julia Lobur, Narosa Publications ISBN81- 7319-609-5.
3. Computer System Architecture‖, John. P.Hayes.

Web References:

1. NPTEL: Computer Science and Engineering - NOC: Computer architecture and organization.

DATABASE MANAGEMENT SYSTEMS

II Year B. Tech. I semester

[Common to CSE, CSE (AI&ML), IT, CSE(Cybersecurity)]

Course Code: 24CT11RC12

L	T	P	C
3	0	0	3

Course Outcomes: At the end of the Course, the student shall be able to

CO1: **Explain** the basic concepts of Database Management System, levels of abstraction, architecture of DBMS. (L2)

CO2: **Construct** the E-R model to perform conceptual design and summarize the concepts of Basic SQL, Relational Algebra, Relational Calculus to manipulate the database. (L3)

CO3: **Apply** Relational Model concepts to perform various operations. (L3)

CO4: **Relate** and plan the concept of data planning and database design using normalization. (L2)

CO5: **Make use of** the ACID properties, concurrency control mechanisms, File Organizations and Indexing Techniques. (L3)

UNIT-I **10 Lectures**

Introduction: Database system, Characteristics (Database Vs File System), Advantages of Database systems, Database Applications. Describing and Storing Data in a DBMS - Relational Model, Levels of abstraction, Data Independence, Structure of a DBMS, People Who Work with Databases.

UNIT-II **10 Lectures**

Introduction to Database Design: Database Design and E-R Diagrams, Entities, Attributes and Entity Sets, Relationship and Relationship Sets, Conceptual Design With the E-R Models.

Basic SQL: Simple Database schema, data types, table definitions (create, alter), Creating tables with relationship, Implementation of key and integrity constraints different DML operations (insert, delete, update), basic SQL querying (select and project) using where clause, arithmetic & logical operations, SQL functions (Date and Time, Numeric, String conversion).

Relational Algebra and Calculus: Relational Algebra-Selection and Projection, Set Operation, Renaming, Joins and Division, Relational Calculus.

UNIT-III **12 Lectures**

Relational Model: Introduction to relational model, concepts of domain, attribute, tuple, relation, importance of null values, The Relational Model Integrity Constraints Over Relations- Key Constraints–Foreign Key Constraints-General Constraints, nested queries, sub queries, grouping, aggregation, ordering, implementation of different types of joins, view (updatable and non-updatable), relational set operations, Complex Integrity Constraints in SQL, Triggers and Active Databases.

UNIT-IV **10 Lectures**

Schema Refinement and Normal Forms: Introduction to schema refinement, functional dependency, Reasoning abouts FDs, Normal forms-1NF, 2NF, 3NF and Boyce-codd normal form (BCNF), Properties of Decomposition- Lossless join and dependency preserving decomposition, Fourth normal form (4NF), Fifth Normal Form (5NF).

UNIT-V **10 Lectures**

Transaction Management and Concurrency Control: Transaction, ACID properties, transaction log and transaction management with SQL using commit rollback and save point, Transactions and Schedules, Concurrent Execution of Transactions, Concurrency control with locking methods-lock granularity, lock types, two phase locking for ensuring serializability, deadlocks, Concurrency control with time stamp ordering- Wait/Die and Wound/Wait Schemes Overview of Storage and Indexing - Data on External Storage, File Organizations and Indexing, Index data Structures.

Text Books:

1. Database Management Systems, 3/e, Raghurama Krishnan, Johannes Gehrke, TMH.
2. Database System Concepts,5/e, Silberschatz, Korth, TMH.

Reference Books:

1. Introduction to Database Systems, 8/e C J Date, PEA.
2. Database Management System, 6/e Ramez Elmasri, Shamkant B. Navathe, PEA.
3. Database Principles Fundamentals of Design Implementation and Management, Corlos Coronel, Steven Morris, Peter Robb, Cengage Learning.

Web References:

1. https://nptel.ac.in/courses/106/105/106105175/

DATA STRUCTURES AND ALGORITHMS

II Year B. Tech. I semester
[CSE (Cybersecurity)]

Course Code: 24CC11RC01

L	T	P	C
3	0	0	3

Course Outcomes: At the end of the Course, the student shall be able to

CO1: **Outline** the concept of ADT, identify suitable data structures to solve problems, and Experiment with different searching and sorting techniques using arrays. (L2, L3)

CO2: **Develop and analyse** algorithms for stacks, queues, and priority queues. (L3, L4)

CO3: **Model** linked list representations for various applications. (L3)

CO4: **Develop and analyse** algorithms for Binary Trees and Binary Search Trees. (L3, L4)

CO5: **Summarize** the concepts of Graph representation, graph traversals, and hashing. (L2)

UNIT-I **12 Lectures**

Data Structures: Definition, Classification of Data Structures, Operation on Data Structures, Abstract Data Type (ADT), Introduction to array ADT. **Algorithms:** Introduction, Algorithm Criteria, Algorithm Specification, Performance Analysis-Space Complexity, Time Complexity, Asymptotic Notations, Amortized Analysis. **Stacks**: Introduction to Stacks, Array Representation of Stacks, Operations on Stacks. **Queues:** Introduction to Queues, Representation of Queues-using Arrays, Operations on Stacks.

UNIT-II **10 Lectures**

Linked List: Introduction, Single linked list, Representation of Linked list in memory, Operations on Single Linked list-Insertion, Deletion, Search and Traversal, Double Linked list, Operations on Double Linked list-Insertion, Deletion, Search and Traversal Circular Linked list.

Heaps: Priority Queue, Definition of a Max Heap, Insertion and Deletion on Max Heap, Heap sort.

Hashing: Introduction, static hashing, hashing functions, overflow handling techniques.

UNIT-III **08 Lectures**

Introduction: Representation of trees

Binary Trees: The ADT, Properties of binary trees, Binary tree Representations, Binary tree Traversals: in-order traversal, pre-order traversal, post-order traversal.

Binary Search Trees: Definition, Searching a BST, Insertion into a BST, Deletion from a BST, Sorting using BST.

UNIT-IV **10 Lectures**

Divide and Conquer: General Method, the master method for solving recurrences, Binary Search, Merge Sort, Quick Sort. **The Greedy Method:** The General Method, Knapsack Problem, minimum-cost spanning trees, Single Source Shortest Paths.

UNIT-V **10 Lectures**

Dynamic Programming: The General Method, All Pairs-Shortest Paths, 0/1 Knapsack Problem.

Backtracking: The General Method, The 8-Queens Problem

Graphs: Representations of Graphs-Adjacency Matrix and using Linked list, Graph Traversals (BFT & DFT).

NP-Hard and NP-Complete problems: The Classes NP-hard and NP-Complete.

Text Books:

1. "Fundamentals of Data Structures in C", 2nd Edition, Horowitz, Sahni, Universities Press.
2. "Fundamentals of Computer Algorithms", 2nd Edition, Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, Universities Press

Reference Books:

1. "Introduction to Algorithms", 3rd Edition by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, PHI.
2. "Data Structures and algorithm analysis in C", 2nded, Mark Allen Weiss.
3. Data Structures Using C. 2nd Edition, ReemaThareja, Oxford.

Web References:

1. https://onlinecourses.swayam2.ac.in/cec25_ma15/preview
2. http://nptel.ac.in/courses/106101060/
3. https://www.programiz.com/dsa?utm

OPERATING SYSTEMS

II Year B. Tech. I semester

[Common to CSE, CSE(AI&ML), IT, CSE(Cybersecurity)]

Course Code: 24CI11RC13

L	T	P	C
3	0	0	3

Course Outcomes: At the end of the Course, the student shall be able to

CO1: **Classify** operating system types, concepts, and core services to understand their role in managing hardware and software resources. (L2)

CO2: **Summarize** the impact of scheduling policies and interprocess communication on system performance in multitasking environments. (L2)

CO3: **Make use of** synchronization mechanisms and deadlock handling strategies to ensure safe and efficient concurrent process execution. (L3)

CO4: **Apply** memory management techniques like paging, segmentation, and virtual memory to optimize resource utilization and system performance. (L3)

CO5: **Utilize** file system structures, allocation methods, and disk scheduling algorithms to enhance data storage and retrieval efficiency. (L3)

UNIT-I **10 Lectures**

Introduction: Operating System's role, Operating-System Structure, Operating-System Operations, Process Management, Memory Management, Storage Management, Protection and Security.

System Structures: Operating-System Services, System Calls, Types of System Calls, Operating System Structure, Operating-System Generation, System Boot.

UNIT-II **12 Lectures**

Process Concept: Process Concept, Process Scheduling, Operations on Processes, Interprocess Communication.

Multithreaded Programming: Overview, Multithreading Models, Threading Issues.

Process Scheduling: Basic Concepts, Scheduling Criteria, Scheduling Algorithms, Thread Scheduling, Multiple Processor Scheduling

UNIT-III **12 Lectures**

Synchronization: The Critical-Section Problem, Peterson's Solution, Synchronization Hardware, Mutex Locks, Semaphores, Classic Problems of Synchronization, Monitors.

Deadlocks: System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock.

UNIT-IV **10 Lectures**

Memory-Management Strategies: Swapping, Contiguous Memory Allocation, Segmentation, Paging, Structure of the Page Table.

Virtual-Memory Management: Demand Paging, Copy-on-Write, Page Replacement, Allocation of Frames, Thrashing.

UNIT-V **08 Lectures**

File System: File Concept, Access Methods, Directory and Disk Structure, File-System Mounting, File Sharing.

Implementing File-Systems: File-System Structure, File-System Implementation, Directory Implementation, Allocation Methods, Free-Space Management.

Mass-Storage Structure: Overview of Mass-Storage Structure, Disk Structure, Disk Scheduling, Swap-Space Management, RAID Structure.

Text Books:

1. Operating System Concepts, Abraham Silberschatz, Peter B. Galvin, Greg Gagne, 9th Edition, Wiley, 2018, ISBN: 978-1118063330.
2. Operating Systems: Internals and Design Principles, William Stallings, 9th Edition, Pearson, 2018, ISBN: 978-1292214290.

Reference Books:

1. Modern Operating Systems, Andrew S. Tanenbaum, Herbert Bos, 5th Edition, Pearson, 2022.
2. Operating Systems: A Concept-Based Approach, Dhananjay M. Dhamdhere, 3rd Edition, McGraw Hill, 2017.
3. Operating Systems: A Design-Oriented Approach, Charles Crowley, 1st Edition, McGraw Hill, 2019.

Web References:

1. https://nptel.ac.in/courses/106106144.
2. https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-828operating system- engineering-fall-2012/.
3. https://cs140.stanford.edu

DATABASE MANAGEMENT SYSTEMS LAB

II Year B. Tech. I semester

[Common to CSE, CSE (AI&ML), IT, CSE(Cybersecurity)]

Course Code: 24CT11RC16

L	T	P	C
0	0	3	1.5

Course Outcomes: At the end of the Course, the student shall be able to

CO1: **Build** the database using SQL DDL, DML commands and make use of built-in-functions to write queries. (L3)

CO2: **Identify** and implement different operators and Apply nested queries to solve real time problems. (L3)

CO3: **Plan** Queries on Joins, Aggregate functions, views. (L3)

CO4: **Construct** simple PL/SQL programs using control statements and exception handling methods. (L3)

CO5: **Develop** applications using PL/SQL including procedures, functions, cursors, packages and triggers. (L3)

List of the Experiments

Implement the following modules using ORACLE

Module-1:

Creation, altering and dropping of tables and inserting rows into a table (use constraints while creating tables) examples using SELECT command.

Module-2:

Queries using Conversion functions (to_char, to_number and to_date), string functions (Concatenation, lpad, rpad, ltrim, rtrim, lower, upper, initcap, length, substr and instr), date functions (Sysdate, next_day, add_months, last_day, months_between, least, greatest, trunc, round, to_char, to_date)

Module-3:

Queries (along with sub-Queries) using ANY, ALL, IN, EXISTS, NOTEXISTS, UNION, INTERSECT Constraints.

Module-4:

Queries using Aggregate functions (COUNT, SUM, AVG, MAX and MIN), GROUP BY, ORDER BY, HAVING clauses

Module-5:

Queries on Views and Joins

Module-6: PL/SQL

1. Create a simple PL/SQL program which includes declaration section, executable section and exception Handling section (Ex. Student marks can be selected from the table and printed for those who secured first class and an exception can be raised if no records were found)
2. Insert data into student table and use COMMIT, ROLLBACK and SAVEPOINT in PL/SQL block.

Module-7:

Develop a program that includes the features NESTED IF, CASE and CASE expression. The program can be extended using the NULLIF and COALESCE functions.

Module-8:

Program development using WHILE LOOPS, numeric FOR LOOPS, nested loops using ERROR Handling, BUILT–IN Exceptions, and USER DEFINED Exceptions, RAISE_APPLICATION _ERROR

Module-9:

Programs development using creation of procedures, passing parameters IN and OUT of Procedures

Module-10:

Program development using creation of functions, invoke functions in SQL Statements and write complex functions

Module-11:

1. Develop programs using features parameters in a CURSOR, FOR UPDATE CURSOR, WHERE CURRENT of clause and CURSOR variables**.**
2. Develop Programs using BEFORE and AFTER Triggers, Row and Statement Triggers and INSTEAD OF Triggers

Case Study:

Select any one application mentioned below and Apply the Database Design steps

Note: A report has to be submitted by every student at the end of the semester

1. Accounting Package for Shops
2. Database Manager for Magazine Agency or Newspaper Agency
3. Ticket Booking for Performances
4. Preparing Greeting Cards & Birthday Cards
5. Personal Accounts - Insurance, Loans, Mortgage Payments, Etc.,
6. Doctor's Diary & Billing System
7. Personal Bank Account
8. Class Marks Management
9. Hostel Accounting
10. Video Tape Library
11. History of Cricket Scores
12. Cable TV Transmission Program Manager
13. Personal Library
14. Sailors Database
15. Suppliers and Parts Database

Reference Books:

1. Oracle: The Complete Reference by Oracle Press.
2. Nilesh Shah, "Database Systems Using Oracle‖, PHI, 2007.
3. Rick F Vander Lans, ―Introduction to SQL‖, Fourth Edition, Pearson Education, 2007.

Web Reference:

1. https://docs.oracle.com/en/
2. https://www.w3schools.com/sql/sql_exercises.asp
3. https://www.w3resource.com/sql-exercises/

DATA STRUCTURES AND ALGORITHMS LAB

II Year B. Tech. I semester

[CSE(Cybersecurity)]

Course Code: 24CC11RC02

L	T	P	C
0	0	3	1.5

Course Outcomes: At the end of the Course, the student shall be able to

CO1: **Apply** different Searching and Sorting Techniques using arrays. (L3)

CO2: **Experiment with** different linear data structure concepts using stacks and Queues. (L3)

CO3: **Develop** linear data structure models using various Linked lists. (L3)

CO4: **Build** Binary Search Tree & AVL tree and examine their traversals. (L3)

CO5: **Apply** DFS and BFS graph traversal techniques. (L3)

List of the Experiments

Implement the following programs with either C/C++/ Java/Python

Module-1: Searching and Sorting

1. Write a program that uses non recursive function to perform linear search to find a value in the given list.
2. Write a program that implement Bubble sort, to sort a given list of integers in ascending order.
3. Write a program that implement Selection sort, to sort a given list of integers in ascending order.
4. Write a program that implement Insertion sort, to sort a given list of integers in ascending order.

Module-2: Stack & Queue

1. Write a program that implement stack (its operations) using arrays.
2. Write a program that implement Queue (its operations) using arrays.

Module-3: Singly Linked List

1. Write a program that uses functions to create and perform operations on singly linked list.

Module-4: Double Linked List

1. Write a program that uses functions to create and perform operations on double linked list.
2. Write a program that uses functions to create and perform operations on circular linked list.

Module-5: Singly Linked List

1. Write a program that uses functions to create and perform operations on singly linked list.

Module-6: Graphs

1. Write a program to implement Depth First Search.
2. Write a program to implement Breadth First Search.

Module-7: **Divide-and-Conquer**

1. Write a program using Divide and conquer paradigm to perform Binary search to find value in a given list.
2. Write a C++ program to demonstrate following sorting techniques
   a) Merge Sort
   b) Quick Sort

   Read at least 15 numbers in a range of 1 to 100 and display the sorted output.

Module-8: The Greedy Method

1. Write a C++ program to find Minimum Cost Spanning Tree of a given connected undirected graph using Prim's algorithm.
2. Write a C++ program to perform Single source shortest path problem for a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm.

Module-9: Dynamic Programming

1. Write a C++ program to implement following algorithms using Dynamic Programming Concept.
   a) All-Pairs Shortest Paths problem using Floyd's algorithm
   b) 0/1 Knapsack problem

Module-10: Graphs

1. Write a C++ program to implement following algorithms using Backtracking Concept.
   a) Sum of subsets problem for a given set of distinct numbers
   b) Hamiltonian cycle problem.

CASE STUDY: Select any one application mentioned below

Note: A report has to be submitted by every student at the end of the semester that includes requirements, design, coding, and output with testing results of a real example.

1. Demonstrate a queue using Linked List and Stack.
2. Model a linked list data structure to add two polynomials.
3. Demonstrate to convert an infix expression into a postfix expression and evaluate to find the result.
4. Design a student record management system that stores student details efficiently and supports fast searching and updating of records by using AVL Trees.
5. A company needs to multiply large sales data matrices quickly. Design an efficient solution to improve the multiplication process with the help of Strassen's Matrix Multiplication.
6. Design a token management system for a bank that continuously serves customers using the available queue space efficiently using Circular Queue.
7. Design a music playlist application that allows users to reverse the order of songs in the playlist using a Singly Linked List.
8. Write a C++ program to implement reliability optimization in system design using Dynamic Programming (DP). The goal is to select a combination of components that provides the maximum system reliability while staying within a given budget.
9. Write a C++ program to implement the Optimal Binary Search Tree (OBST) algorithm using Dynamic Programming (DP). The goal is to arrange search terms based on their frequencies so that frequent queries are retrieved faster, optimizing the auto-complete feature in a search engine.
10. Write a C++ program to solve the 0/1 Knapsack Problem using the Branch and Bound approach. Consider a scenario where a relief truck has a fixed weight capacity, and you must select essential supplies (food, medicine, blankets, etc.) to maximize the total priority value while staying within the weight limit.
11. A supermarket stores the product IDs of items sold each day in an unsorted list. The manager wants to identify the fastest sorting technique for generating sales reports. Sort the given product IDs using Radix Sort, Counting Sort, and Cycle Sort, compare their execution time and memory

usage, and recommend the most suitable algorithm for this application.

12. A library stores book IDs in a sorted list and needs to quickly locate a requested book. Search for the given book ID using Linear Search, Binary Search, and Interpolation Search, compare their search time and number of comparisons, and recommend the most suitable algorithm for this application.

Text Books:

1. Reema Thareja, Data Structures Using C, 2nd Edition, Oxford University Press, 2014.
2. Ellis Horowitz, Sartaj Sahni, and Susan Anderson-Freed, Fundamentals of Data Structures in C, 2nd Edition, Universities Press, 2008.

Reference Books:

1. Data Structures: A PseudoCode Approach, 2/e, Richard F.Gilberg, Behrouz A. Forouzon, Cengage.
2. Richard F. Gilberg and Behrouz A. Forouzan, Data Structures: A Pseudocode Approach with C, 2nd Edition, Cengage Learning, 2005.
3. Mark Allen Weiss, Data Structures and Algorithm Analysis in C, 2nd Edition, Pearson Education, 1997.

Web References (e-Resources):

1. https://onlinecourses.swayam2.ac.in/cec25_ma15/preview
2. https://nptel.ac.in/courses/106101060
3. https://www.programiz.com/dsa

OPERATING SYSTEMS LAB

II Year B. Tech. I semester

[Common to CSE, CSE(AI&ML), IT, CSE(Cybersecurity)]

Course Code: 24CT11RC17

L	T	P	C
0	0	3	1.5

Course Outcomes: At the end of the Course, the student shall be able to

CO1: **Make use of** Linux commands for shell programming. (L3)

CO2: **Construct** programs for various OS concepts and Linux commands using System Calls. (L3)

CO3: **Develop** and implement programs to demonstrate various CPU Scheduling Algorithms, Process Synchronization Concepts in Operating Systems and Deadlock Prevention and Avoidance Techniques. (L3)

CO4: **Build** programs for Memory Management Techniques and Page Replacement Algorithms. (L3)

CO5: **Develop** programs for various DISK Scheduling Algorithms. (L3)

List of the Experiments

Module -1: Implementation of basic Linux commands

1. Navigate to a directory, create a subdirectory, copy files into it, change permissions, and list them using various commands.
2. Create three empty text files using the touch command (e.g., file1.txt, file2.txt, file3.txt).
3. Use the cp command to copy one of the text files to another directory. If no other directory exists, create one using mkdir and then copy the file. Rename one of the text files using the mv command.
4. Create a text file in Vi that contains a list of files in your home directory. Include descriptions for each file. Use commands like :w, :q, :set number, and navigation commands to format it.
5. Modify the file using dd, yy, p, and /search commands. Save the file and quit.

Module -2: Write a C program on scheduling algorithms in Linux/Windows OS.

1. Implementation and Analysis of FCFS Scheduling Algorithm.
2. Implementation and Analysis of SJF Scheduling Algorithm.

3. Implementation and Analysis of Round Robin Scheduling Algorithm.

Module 3: Implementation of Critical Section Problem Using Peterson's Algorithm

1. Mutual exclusion
2. Progress
3. Bounded waiting between two processes.

Module 4: Write simple shell programming in Linux OS.

1. Hello World Script
2. List Files in Directory
3. Check if a File Exists
4. Count Lines in a File
5. Sum of Numbers
6. Display Date and Time
7. Reverse a String
8. Calculate Factorial
9. Check Prime Number

Module 5: Shell programming using decision making constructs, loop constructs, file and directory manipulation

1. **Decision-Making Constructs:** Write a shell script that checks whether a number entered by the user is positive, negative, or zero. Use if, else, and elif statements for the decision making.
2. **Loop Constructs:** Write a shell script that prints the first 10 Fibonacci numbers using for loop.
3. **File and Directory Manipulation:** Write a shell script that accepts a file name from the user and checks whether it exists. If the file exists, display the number of lines in the file; otherwise, print "File does not exist."

Module-6: Simple C programs using command line arguments, system calls, library function calls, make utility in Linux OS.

1. **Command-Line Arguments: Simple Calculator: Description**
2. Write a C program that accepts command-line arguments to perform basic arithmetic operations (addition, subtraction, multiplication, division). The user should provide the operation type (+, -, *, /) and two operands.

3. **System Calls: File Copy Using open(), read(), and write()**
4. Write a C program that should copy a file from one location to another using system calls like open(), read(), and write().
5. **Library Function Calls: String Reversal**
6. Write a C program that takes a string as input and returns the reversed string using standard library functions.

Module -7

1. Write C programs to study deadlock avoidance and detection in Linux/Windows OS.

Module-8: C programs using system call to create processes and study parent, child process mechanism in Linux OS.

1. Simple Parent-Child Process using fork().
2. Parent Waiting for Child to Exit using wait().

Module-9: Write a C Programs to simulate free space management in Linux/Windows OS.

a) first fit b) best fit c) worst fit

Module-10: Write a C programs to create process chaining, spawning in Linux OS.

1. Process Chaining using fork()
2. Process Spawning using fork() and exec()

Module-11: Write a C programs to error handling using errno(), perror() function in Linux OS.

1. File Opening Error
2. Division by Zero
3. Memory Allocation Error
4. Invalid File Descriptor

Module-12: Write C programs to use pipe system call for interprocess communication between Parent and Child Process in Linux OS.

Module-13: Write C programs for page replacement implementing in Linux/Windows OS.

a)FIFO b) Optimal c)LRU

Module-14: Write C programs for disk scheduling algorithms in Linux/Windows OS.

a) SCAN b) SSTF c) LOOK

Case study: Select any one case study mentioned below.

Note: At the end of the semester, each student is required to submit a report. Compare any of two operating systems such as Windows, Linux, macOS, Android, or others.

The report should include a case study covering the following topics:

1. Different operating systems implement their file systems in unique ways, depending on their architecture and design objectives. Write an overview of how file systems function across various operating systems.
2. Memory management is responsible for allocating and deallocating memory space to various programs and ensuring efficient utilization of system memory.
3. Different operating systems implement scheduling in unique ways based on their design goals and hardware requirements. Below is an overview of scheduling across various operating systems
4. Protection ensures the integrity, confidentiality, and availability of the system, preventing unintentional or intentional harm, an overview of how protection is implemented across different operating systems.
5. Each operating system handles I/O operations through a combination of system calls, device drivers, and I/O subsystems to provide seamless interaction with hardware, an overview of how different operating systems manage I/O operations input/output.

References Books:

1. Unix concepts and applications by Sumitabha Das, TMH Publications.
2. Unix programming by Stevens, Pearson Education.
3. Shell programming by Yashwanth Kanetkar.
4. Operating System Concepts by Silberschatz, and Peter Galvin.

Web References(e-Resources):

1. https://nptel.ac.in/courses/106106144
2. https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-828-operating- system-engineering-fall-2012/
3. https://cs140.stanford.edu

WEB APPLICATION SECURITY LAB

II Year B. Tech. I semester

[Skill Course: CSE(Cybersecurity)]

Course Code: 24CC11SC01

L	T	P	C
1	0	2	2

Course Outcomes: At the end of the Course, the student shall be able to

CO1: Design secure web inputs by implementing strict boundary validation and parameter sanitization to neutralize injection and tampering vulnerabilities.

CO2: Evaluate live web traffic and data encapsulation layers to identify cleartext transmission risks and confirm SSL/TLS encryption.

CO3: Construct secure Python and Express.js backend services featuring structured CRUD operations, rate-limiting, and safe error-handling middleware.

CO4: Defend database stores and authentication workflows against exploitation by deploying parameterized queries and non-verbose security messaging.

CO5: Audit web application deployments using automated scanning tools to discover, analyze, and mitigate severe vulnerability types.

List of the Experiments

Module -1: Web Application Security Fundamentals

1. Write a program to implement basic *input handling* for common Web Application functions.
2. Develop a server-side program to *enforce Boundary Validation on incoming user inputs* using an input validation library.

Software Requirements: VSCode, Node, postman

Module -2: Live Web / Network Traffic Analysis

1. Install Wireshark and explore the various protocols by capturing Live *Web traffic* / network interface or to inspect raw frame components, data encapsulation layers, and protocol headers.
2. Analyze the difference between *HTTP vs HTTPS* packet layers by capturing active login credentials over cleartext versus SSL/TLS encrypted application streams.

Software Requirements: Wireshark, Npcap

Module 3: Attacking Data Stores using SQL injection Techniques

Write a program to demonstrate *Bypassing a Login Web Pages* using SQL Injection into interpreted contexts, and implement parameterized queries to *prevent it.*

Software Requirements: XAMPP, PHP, MySQL, Apache Server, Web Browser

Module 4: Upstream Proxy Configuration and Parameter Tampering using BurpSuite

1. Configure Burp Suite as a local upstream proxy to intercept and decrypt live browser HTTP/S request-response parameters in Real time.
2. Trap an active submission request to manually locate, modify, and fuzz an internal transaction or user role variable before forwarding the payload to the server.

Software Requirements: Burp Suite, Java, OWASP Juice Shop, Node.js, Angular, Express, REST APIs

Module 5: Authentication Security

1. Create simple REST API using Python to handle secure record retrieval via GET and Data resource initialization via POST payloads.

Software Requirements: Python, Flask, Postman, VS Code

Module-6: Vulnerability Assessment

1. Identify the vulnerabilities using OWASP ZAP tool by deploying its active and passive scanning components against a targeted local web application deployment. or use http://vulnweb.com/

Software Requirements: OWASP ZAP, Java, DVWA, OWASP Juice Shop, WebGoat, XAMPP

Module -7: Attacking and Securing Authentication Mechanisms

1. Write a program to *restrict Brute Forcible Login attempts* using rate-limiting middleware and prevent user enumeration by eliminating Verbose Failure Messages.
2. Attack the website using Social Engineering method

Software Requirements: Python, Flask

Module-8 Analysis and Exploitation of Cross-Site Scripting (XSS)

1. Write a program *to simulate different Varieties of XSS* (Cross Site Scripting) by injecting malicious scripts into vulnerable frontend input fields.

Software Requirements: Python, Flask

Case Studies:

Select any one practical application mentioned below

Note: A **report** has to be submitted by every student at the end of the semester that includes design, coding, output, etc.

Case Study 1 (Module 1): E-Commerce Product Quantity Validator

- **Scenario:** A retail website's order checkout form accepts quantities without verification. Attackers can input negative numbers or massive strings, forcing the database to corrupt inventory counts or crash server memory.
- **Deliverables:**
    1. Write a server-side routine using an input validation library to restrict item quantities strictly to integers between 1 and 100.

Case Study 2 (Module 2): Corporate Login Packet Audit

- **Scenario:** An employee connects to a legacy company portal from a public network. A network auditor flags the connection because credentials might be passing in plaintext, leaving them open to local sniffing.
- **Deliverables:**
    1. Configure a Wireshark capture rule to isolate HTTP POST traffic and pinpoint exact form variables displaying plaintext usernames and passwords.
    2. Migrate the application route to HTTPS and capture the traffic again to verify that all credential frames are hidden inside an encrypted TLS envelope.

Case Study 3 (Module 3): Medical Record Search Extraction (SQLi)

- **Scenario:** A hospital's patient record lookup search bar strings together raw text inputs directly into an active SQL command. An attacker uses a ' UNION SELECT trick to map out other database tables, leaking private doctor credentials on the screen.
- **Deliverables:**
    1. Replicate a login bypass flaw by crafting a text injection script that manipulates the SQL command interpreter to drop authentication blocks.
    2. Secure the entire search route by converting the raw database script into safe, parameterized queries that treat inputs strictly as data.

Case Study 5 (Module 5): Secure REST API Patient Database

- **Scenario:** A clinic needs a lightweight data layer to handle patient information but requires clear endpoint segregation so that account management features do not conflict or accidentally expose records.
- **Deliverables:**
    1. Code a Python-based REST API (using Flask or FastAPI) that securely executes data retrieval queries via GET and inputs new assets using POST.
    2. Complete the API's CRUD routing structure by implementing safe state-overwrites via PUSH and robust authorization controls on DELETE pathways.

Case Study 6 (Module 6): Automated FinTech Scan & Error Shielding

- **Scenario:** A banking portal's testing server crashes whenever a bad request is received, printing raw database lines and internal file directory paths on the user's screen.
- **Deliverables:**
    1. Deploy OWASP ZAP's active and passive scanning tools against the application to locate, catalog, and grade baseline structural vulnerabilities.
    2. Build an Express.js error-handling middleware that catches application crashes and displays generic, friendly error messages without leaking stack traces.

Case Study 7 (Module 7): Portal Brute-Force and Credential Reset Audit

- **Scenario:** An automation bot attacks a corporate login page by running millions of guesses. The site tells the bot exactly when a username is real (e.g., *"Password incorrect"* vs. *"User not found"*), allowing the bot to discover valid emails.
- **Deliverables:**
    1. Deploy a rate-limiting middleware block to drop automated login attempts after 5 failures and replace descriptive errors with generic messages.
    2. Develop a secure password recovery routine that generates random, unguessable single-use tokens instead of predictable time-based values.

Case Study 8 (Module 8): Social Feedback Board (XSS Trap)

- **Scenario:** A public guestbook allows visitors to leave text reviews. A prankster posts a review containing an active <script> tag that executes a cookie-stealing routine in the browser of

anyone who views the guestbook.

- **Deliverables:**
    1. Build a vulnerable feedback form to show how stored and reflected script strings can execute unauthorized commands inside a client's browser.
    2. Create a functional exploit workflow that captures session cookies, and implement context-aware output encoding to neutralize script tags.

Text Books:

1. The Web Application Hacker's Handbook: Finding and Exploiting Security Defydd Stuttard, Marcus Pinto Wiley Publishing, Second Edition.
2. Andrew Hoffman, Web Application Security: Exploitation and Countermeasures for Modern Web Applications, O'Reilly Media, 2020.

Reference Books:

1. “Professional Pen Testing for Web application”, Andres Andreu, Wrox Press
2. “Web Application Security” Carlos Serrao, Vicente Aguilera, Fabio Cerullo, Springer; 1st Edition
3. Michal Zalewski, The Tangled Web: A Guide to Securing Modern Web Applications, No Starch Press, 2011.

Web References:

1. https://owasp.org/
2. https://portswigger.net/web-security
3. https://www.nist.gov/cybersecurity

ENVIRONMENT SCIENCE

II Year B. Tech. I Semester

[Common to CSE, IT]

L	T	P	C
2	0	0	0

Course Code: 24BC11MC01

Course Outcomes: At the end of the Course, the student shall be able to

CO1: Understanding the impacts of developmental activities and mitigation measures for sustainable environment.

CO2: Categorize the importance of natural resources management for the sustenance of the life and the society.

CO3: Differentiate various forms of pollutions and their impact on the environment

CO4: Discuss elements of Sustainable Development, energy and environmental management.

CO5 : Develop technologies on the basis of ecological principles and environmental regulations which in turn helps in sustainable development

UNIT-I **10 Lectures**

Introduction: Structure and functions of Ecosystems-Ecosystems and its Dynamics-Value of Biodiversity-impact of loss of biodiversity, Conservation of bio-diversity. Environmental indicators - Global environmental issues and their impact on the ecosystems. Salient features of international conventions on Environment: Montreal Protocol, Kyoto protocol, Ramsar Convention on Wetlands, Stockholm Convention on Persistent Organic Pollutants, United Nations Framework Convention on Climate Change (UNFCCC)

UNIT-II **10 Lectures**

Natural Resources Management: Importance of natural resources management-Land as resource, Land degradation, Soil erosion and desertification, Effects of usage of fertilizer, herbicides and pesticide-watershed management. **Forest resources:** Use and over-exploitation, Mining and dams – their effects on forest ecosystems and the living beings. **Water resources:** Exploitation of surface and groundwater, Floods, droughts, Dams: benefits a nd costs. **Mineral Resources**: Impact of mining on the environment and possible environmental management options in mining and processing of the minerals. Sustainable resource management (land, water, and energy), and resilient design under the changing environment.

UNIT-III **10 Lectures**

Environmental Pollution: Local and Global Issues. Causes, effects and control measures. Engineering aspects of environmental pollution control systems. Air pollution: impacts of ambient and indoor air pollution on human health. Water pollution: impacts water pollution on human health and loss of fresh water resources. Soil pollution and its impact on environment, Marine pollution and its impact on blue economy. Noise pollution. Air (prevention and control of pollution) Act, Water (prevention and control of pollution) Act and their amendments. Salient features of Environmental protection Act, 1986. Pollution control technologies: Wastewater Treatment methods: Primary, secondary and Tertiary.

UNIT-IV **10 Lectures**

Sustainable Development: Population and its explosion Fundamentals of Sustainable Development–Sustainability Strategies and Barriers – Industrialization and sustainable development. Circular economy concepts in waste (solid and fluid) management. **Energy and Environment:** Environmental Benefits and challenges, Availability and need of conventional energy resources, major environmental problems related to the conventional energy resources, future possibilities of energy need and availability. **Renewable Energy:** Production of Hydrogen via Water Splitting Using Photocatalytic and Photo electrocatalytic Route process of photovoltaic energy conversion, solar energy conversion technologies and devices, their principles, working and applications, disposal of solar panel after their usage. **Biomass energy:** Concept of biomass energy utilization, types of biomass energy, conversion processes, Wind Energy, energy conversion technologies, their principles, equipment and suitability in context of India.

UNIT-V **10 Lectures**

Solid waste management: Important elements in solid waste management- Waste to energy concepts. **Management of plastic waste and E-waste:** Sources, generation and characteristics of various e- and plastic wastes generated from various industrial and commercial activities; Waste management practices including onsite handling, storage, collection and transfer. E-waste and plastic waste processing alternatives. E-Waste management rules and Plastic waste management rules, 2016 and their subsequent amendments.

Text Books:

1. Environmental Studies by R. Rajagopalan, Oxford University Press

2. Bharucha, Erach (2004). Textbook for Environmental Studies for Undergraduate Courses of all Branches of Higher Education, University Grants Commission, New Delhi.

3. Base, M., Xavier, S. (2016). Fundamentals of Environmental Studies, Cambridge University Press, India 67

Reference Books:

1. Sharma, P. D., & Sharma, P. D. (2005). Ecology and environment. Rastogi Publications

2. Agarwal, K.C. 2001 Environmental Biology, Nidi Publ. Ltd. Bikaner.

3. Clark R.S. (2001). Marine Pollution, Clanderson Press Oxford (TB) 2018.