



SIDDHARTHA
ACADEMY OF HIGHER EDUCATION

An Institution **Deemed to be University**

(Under Section 3 of UGC Act, 1956)

Kanuru, Vijayawada - 520 007, AP. www.vrsiddhartha.ac.in

91 866 2582333

866 2582334

866 2584930

V R SIDDHARTHA SCHOOL OF ENGINEERING

B. Tech
in
INFORMATION TECHNOLOGY

(B. Tech. IT Programme Accredited by NBA)

SEMESTER III (II/IV)



Scheme of Instruction and Syllabus

w.e.f. 2025-26

SIDDHARTHA ACADEMY OF HIGHER EDUCATION
(Deemed to be University)
V R SIDDHARTHA SCHOOL OF ENGINEERING
DEPARTMENT OF INFORMATION TECHNOLOGY
SCHEME OF INSTRUCTIONS – SU24

Semester III

Contact Hours: 28

S.No	Course Code	Course Category	Course Name	L	T	P	Credits
1	24MA201	BS	Probability & Statistics	4	0	0	4
2	24IT201	ES	Advanced Data Structures and Algorithms	3	0	0	3
3	24IT202	PC	Operating Systems	2	0	2	3
4	24IT203	PC	Computer Organization and Architecture	3	0	0	3
5	24IT204	PC	Object Oriented Programming through C++	3	0	0	3
6	24IT281	PL	Object Oriented Programming through C++ Lab	0	0	3	1.5
7	24IT282	ES	Advanced Data Structures Lab	0	0	3	1.5
8	24IT283	PL	Web Programming	0	0	2	1
9	24UC201	MC	Universal Human Values-II	3	0	0	3
Total				18		10	23

Legend:

L: Lecture Hours, **T:** Tutorial Hours, **P:** Practical Hours

BS: Basic Sciences, **ES:** Engineering Sciences, **PC:** Professional Core, **PL:** Programming Lab, **MC:** Mandatory Course

24MA201 - PROBABILITY & STATISTICS (AI & DS, CSE, CSE (AI & ML), IT)			
Course Category:	Basic Sciences (BS)	Credits:	4
Course Type:	Theory	Lecture-Tutorial-Practice:	4-0-0
Prerequisites:	10+2 Mathematics	Continuous Evaluation:	40
		Semester End Evaluation:	60
		Total Marks:	100

Course Description

An overview of the random variables, discrete & continuous probability distributions, Inference concerning means, variance, proportions and non parametric tests with a focus on their applications in solving engineering problems

Course Objectives

- Fundamentals of probability and cover discrete probability distributions such as Binomial and Poisson.
- Teach continuous random variables and continuous probability distributions, including Normal, Normal approximation and Joint probability distributions.
- Explain the concepts of sample and population, point and interval estimations, and hypothesis testing for one and two means.
- Discuss the estimation of variances, Proportions, Hypothesis testing for one and two variances, as well as proportions, r x c analysis.
- Introduce and discuss about Non Parametric tests like The Sign Test, Rank-Sum tests, Correlation based on ranks, Tests of Randomness.

Course Outcomes

At the end of the course, the student will be able to:

- CO1:** Apply discrete random variables to determine probability distributions like Binomial and Poisson for decision-making [K3]
- CO2:** Apply continuous random variables to the normal distribution, approximate the binomial distribution using the normal distribution, and joint probability distributions [K3]
- CO3:** Implement point and interval estimation methods to estimate population parameters and test hypotheses concerning the one and two means [K3]
- CO4:** Utilize interval estimation to estimate variances and proportions, testing of hypotheses concerning variances and proportions [K4]
- CO5:** Apply some Non parametric tests and assess correlation based on ranks as well as test of randomness. [K3]

Mapping of Course Outcomes to Program Outcomes

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

Course Content

Unit - I: Probability Distributions (Discrete)

Random Variables, Binomial distribution, Hyper Geometric Distribution, Poisson approximation to the Binomial distribution, Poisson process.

Description: This unit introduces students to the concept of discrete random variable and discrete probability distributions Binomial and Poisson for decision making.

Learning Outcome: Students will be

- able to determine mean and variance of discrete random variables
- able to find the probability with respect to binomial and Poisson distributions.

Web Resources:

1. <https://www.youtube.com/watch?v=UftY0e2ilM4>

Unit II: Probability Distributions (Continuous) & Joint Probability:

Probability Densities: Continuous random variables, Normal distribution, Normal approximation to the Binomial distribution.

Joint distribution: Joint Distributions-Discrete and Continuous.

Description:

This unit covers continuous random variable and its properties, normal probability distribution and joint probability distributions.

Learning Outcome: Students will be

- able to find statistical parameters mean, variance and standard deviation in case of continuous random variables
- able to determine distribution function using density function and vice versa of normal probability distribution and determine joint probability distributions.

Web Resources:

1. <http://acl.digimat.in/nptel/courses/video/106105239/L12.html>

Unit III:Hypotheses Concerning Mean

Inferences Concerning a Mean: Point Estimation, Interval Estimation, Tests of Hypotheses, Null Hypotheses and Tests of Hypotheses, Hypotheses concerning one mean, Relation between tests and confidence intervals, Comparisons-Two independent large samples, Comparisons-Two independent small samples.

Description: This unit focuses on the concepts of sample, population and estimation of population mean and testing the hypothesis concerning one and two means of population

Learning Outcome: Students will be

- able to find the confidence interval, maximum error
- able to apply the testing of hypothesis concerning mean to take decision regarding estimated population mean.

Web Resources:

1. <https://www.youtube.com/watch?v=IEP3swFeauE>

Unit IV:Hypotheses Concerning Variances and Proportions

Inferences Concerning Variances: Estimation of variances, Hypotheses concerning one variance, Hypotheses concerning two variances.

Inferences Concerning Proportions: Estimation of Proportions, Hypotheses concerning one Proportion, Hypotheses concerning several Proportions, The analysis of $r \times c$ tables.

Description: This unit introduces testing of hypothesis concerning one & two variances, proportions.

Learning Outcome: Students will be

- able to evaluate confidence interval and maximum error.
- apply several proportions, $r \times c$ analysis to take decisions various engineering and social problems.

Web Resources:

1. <https://archive.nptel.ac.in/courses/103/106/103106120/>

Unit V:Non Parametric Tests

Non parametric Tests: Introduction, The Sign Test, Rank-Sum tests, Correlation based on ranks, Tests of Randomness.

Description: This unit covers non parametric tests like The Sign Test, Rank-Sum tests, Correlation based on ranks, Tests of Randomness.

Learning Outcome: Students will be

- able to understand to attain the best possible precision when the data is doubtful whether the assumption of normality can be met.

Web Resources:

1. <https://archive.nptel.ac.in/courses/103/106/103106120/>

TEXT BOOK(S)

1. Probability and Statistics for Engineers, Eighth edition by Richard A. Johnson
Prentice Hall of India.

REFERENCE BOOKS

1. Probability & Statistics for Engineers & Scientist by R.E. Walpole, R.H.Myers
&S.L.Myers, Sixth Edition, Prentice Hall of India / Pearson Education.
2. Probability and Statistics, Purna Chandra Biswal, Pearson Education Prentice Hall
of India 2007.
3. Probability and Statistics by T.K.V.Iyengar, B.Krishna Gandhi, S.Ranganatham,
M.V.S.S.N.Prasad S.Chand.

24IT201-ADVANCED DATA STRUCTURES & ALGORITHMS			
Course Category:	ES	Credits:	3
Course Type:	Theory	Lecture-Tutorial-Practice:	3-0-0
Prerequisites:	Data Structures Programming Fundamentals	Continuous Evaluation:	40
		Semester End Evaluation:	60
		Total Marks:	100

Course Description

This course introduces fundamental concepts of algorithm design and analysis. It covers various algorithmic paradigms, including Divide and Conquer, Greedy Method, Dynamic Programming, Backtracking, and Branch and Bound. Additionally, it explores advanced data structures and the classification of computational problems as NP-Hard and NP-Complete.

Course Objectives

- Understand algorithm design principles and their efficiency analysis.
- Learn different algorithmic techniques and their applications.
- Analyze computational complexity using asymptotic notation.
- Implement algorithms for real-world problem-solving.
- Recognize NP-complete and NP-hard problems and their significance

Course Outcomes

At the end of the course, the student will be able to

- CO1:** Analyze the efficiency of algorithms using time and space complexity. [K4]
- CO2:** Implement algorithmic techniques like Divide and Conquer, Greedy, and Dynamic Programming. [K3]
- CO3:** Solve optimization problems using backtracking and branch-and-bound techniques. [K3]
- CO4:** Apply advanced tree data structures for efficient searching and sorting. [K3]
- CO5:** Differentiate between P, NP, NP-Complete, and NP-Hard problems. [K2]

Mapping of Course Outcomes to Program Outcomes

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

Course Content

Unit I: Introduction & Divide and Conquer (1-2)

Introduction to Algorithms: Definition, Algorithm Specification, Pseudo-code Conventions, Performance Analysis: Space Complexity, Time Complexity, Asymptotic Notation: Big-O, Omega, Theta, Little-o.

Divide and Conquer: General method, Binary Search, Finding Maximum and Minimum, Sorting Algorithms: Merge Sort, Quick Sort, Matrix Multiplication: Strassen's Algorithm.

Description: This unit introduces the concept of algorithm efficiency, asymptotic notation, and the Divide and Conquer approach. Students will learn to analyze algorithm performance and implement fundamental algorithms using this technique.

Learning Outcome: Students will understand algorithm efficiency, asymptotic notation, and the Divide and Conquer paradigm.

Web Resources:

1. **NPTEL:** Introduction to Algorithms
2. **GeeksforGeeks:** Divide and Conquer Algorithms

Unit II: Greedy Method (3-5)

Greedy Method: General method, Knapsack Problem, Job Sequencing with Deadlines, Minimum Cost Spanning Trees: Prim's and Kruskal's Algorithms, Shortest Path Problem: Single-Source Shortest Path.

Description: This unit covers the Greedy method for problem-solving, explaining its applications in optimization problems such as Minimum Spanning Trees and Shortest Paths.

Learning Outcome: Students will apply Greedy strategies to optimization problems and understand their efficiency.

Web Resources:

1. **GeeksforGeeks:** Greedy Algorithms
2. **MIT OpenCourseWare:** Algorithmic Thinking

Unit III: Dynamic Programming (6-8)

Dynamic Programming: General method, All-Pairs Shortest Path Problem, Single-Source Shortest Paths with General Weights, String Editing, 0/1 Knapsack Problem, Travelling Salesperson Problem.

Description: This unit focuses on the Dynamic Programming approach, which is used to solve problems by breaking them into smaller subproblems. Applications include shortest paths, sequence alignment, and optimization problems.

Learning Outcome: Students will understand how Dynamic Programming optimizes recursive solutions.

Web Resources:

1. **GeeksforGeeks:** Dynamic Programming
2. **Coursera:** Advanced Algorithms

Unit IV: Backtracking & Branch-and-Bound(9-11)

Backtracking: General method, 8-Queens Problem, Sum of Subsets, Graph Coloring, Hamiltonian Cycles.

Branch and Bound: General Method, 0/1 Knapsack Problem, Travelling Salesperson Problem (TSP) using Branch and Bound.

Description: This unit introduces Backtracking and Branch and Bound, two techniques used for solving combinatorial and optimization problems. Applications include constraint satisfaction and game solving.

Learning Outcome: Students will explore constraint-based problem-solving techniques.

Web Resources:

1. **GeeksforGeeks:** Backtracking Algorithms
2. **NPTEL:** Branch and Bound

Unit V: Advanced Data Structures & NP-Completeness (12-14)

Advanced Data Structures: AVL Trees: Creation, Insertion, Deletion, Applications; B-Trees: Creation, Insertion, Deletion, Applications; Red-Black Trees: Properties, Searching, Insertion, Deletions, Top-down operations; Heap Trees (Priority Queues): Min and Max Heaps, Operations, Applications.

NP-Hard and NP-Complete Problems: Concepts, Non-Deterministic Algorithms, Classification.

Description: This unit covers advanced data structures like AVL Trees, B-Trees, Red-Black Trees, and Heaps. Additionally, it introduces NP-completeness, exploring problems that are computationally hard.

Learning Outcome: Students will learn advanced tree structures and explore computational complexity theory.

Web Resources:

1. **GeeksforGeeks:** Red-Black Trees
2. **MIT OpenCourseWare:** Computational Complexity

TEXT BOOK(S)

1. Ellis Horowitz, Sartaj Sahni, Dinesh Mehta, *FUNDAMENTALS OF DATA STRUCTURES in C++*, 2nd edition, University Press.
2. Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekharan, *FUNDAMENTALS OF COMPUTER ALGORITHMS*, 2nd edition, University Press.

REFERENCE BOOKS

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, *Introduction to Algorithms*, PHI Learning Pvt. Ltd., New Delhi, 2010.
2. Lee, Kent D., Hubbard, Steve, *Data Structures and Algorithms with Python*, 1st edition, Springer International Publishing, 2015.

E-RESOURCES AND OTHER DIGITAL MATERIAL

1. Sudarshan Iyengar, Assistant Professor, CSE Department, IIT Ropar, *Programming, Data Structures and Algorithms* [NPTEL], (26 May, 2021). Available: <https://nptel.ac.in/noc/courses/noc18/SEM1/noc18-cs25/>
2. Erik Demaine, Professor of Computer Science at the Massachusetts Institute of Technology, *Advanced Data Structures* [MIT OpenCourseWare], (26 May, 2021). Available: <http://ocw.mit.edu/>
3. https://www.tutorialpoint.com/advanced_data_structures/index.asp

4. <http://peterindia.net/Algorithms.html>

24IT202 - OPERATING SYSTEMS			
Course Category:	Programme Core	Credits:	3
Course Type:	Theory	Lecture-Tutorial-Practice:	2-0-2
Prerequisites:	Programming Fundamentals	Continuous Evaluation:	40
		Semester End Evaluation:	60
		Total Marks:	100

Course Description

This course provides a comprehensive understanding of operating systems, covering fundamental concepts such as system structure, process management, scheduling, synchronization, deadlocks, memory management, and file systems. It explores multithreading, virtual memory, disk management, and access control, equipping students with essential knowledge for efficient system design and resource management.

Course Objectives

- Develop a clear understanding of operating system concepts, including system structure, process management, and synchronization.
- Learn to design efficient scheduling algorithms for process execution and resource allocation.
- Explore memory management techniques such as paging, segmentation, and virtual memory.
- Understand and apply methods for deadlock prevention, avoidance, and detection.
- Analyze file system management and disk scheduling to improve system performance.

Course Outcomes

At the end of the course, the student will be able to:

- CO1:** Describe the fundamental concepts of operating system services, processes, multithreading, file systems, directory structures, and RAID. [K2]
- CO2:** Implement page replacement, CPU scheduling, and disk scheduling algorithms to enhance system performance. [K3]
- CO3:** Analyze and apply appropriate techniques for memory management, CPU scheduling, and disk scheduling to solve system-related challenges. [K4]
- CO4:** Identify and evaluate file allocation methods and free space management for efficient data handling. [K4]
- CO5:** Examine and assess synchronization mechanisms and deadlock handling techniques to effectively manage concurrency and resource allocation in operating systems. [K4]

Mapping of Course Outcomes to Program Outcomes

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

Course Content

Unit - I: Introduction (1-3)

Operating System: Operating System - User View, System View, Operating-System Structure, Operating System Operations, Operating-System Services, System Calls.

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

Threads: Overview, Multicore Programming, Multi-Threading Models, Threading Issues.

Description:

This unit introduces the fundamental concepts of operating systems, providing insights into their structure, operations, and services from both user and system perspectives. It covers essential process management topics, including process scheduling, inter-process communication, and system calls. Additionally, it explores multithreading, focusing on multicore programming, threading models, and concurrency issues.

Exercises:

1. List the basic LINUX commands
2. Write a program to demonstrate the creation of a child process using fork().
3. Write a program to demonstrate the creation of a child process using fork(). The parent and child processes should print their process IDs.
4. Write a program to create a new process using fork() and ensure both parent and child processes execute a loop to print numbers from 1 to 5.
5. Write a program where the parent process waits for the child to finish before printing its message.
6. Demonstrate a zombie process using fork(). The parent process should not wait for the child to finish.

Learning Outcome: Students will be able to understand operating system fundamentals, process management, and multithreading concepts.

Web Resources:

1. Computer Science and Engineering - Noc: Operating System Fundamentals. [Online] <https://archive.nptel.ac.in/courses/106/105/106105214/>
2. GeeksforGeeks - Data Structures Basics [Online] <https://www.geeksforgeeks.org/operatingsystems/>
3. Tutorialspoint Operating System: [Online] https://www.tutorialspoint.com/operating_system/index.htm

Unit II Process Scheduling and Synchronization (4-6)

Process Scheduling: Basic Concepts, Scheduling Criteria, Scheduling Algorithms

Synchronization: Background, The Critical-Section Problem, Petersons Solution, Synchronization Hardware, Mutex Locks, Semaphores, Classic Problems of Synchronization.

Description:

This unit introduces process scheduling concepts, including scheduling criteria and algorithms for efficient CPU utilization. It also covers synchronization mechanisms, addressing concurrency challenges through critical-section solutions, mutex locks, semaphores, and classic synchronization problems to ensure process coordination and data consistency.

Exercises:

1. Implement SJF and Round Robin Algorithms (QS:2ms) by finding Avg. Waiting Time, Turnaround Time Etc.

Process	Arrival Time	Burst Time (ms)
P1	0	5
P2	1	3
P3	2	8
P4	3	6

Table 1: Process Scheduling Table

2. For the above Processes calculate the average turnaround time for these processes with the FCFS scheduling algorithm
3. For the above Processes calculate the average turnaround time for these processes with the SJF scheduling algorithm.
4. For the above Processes calculate the average turnaround time for these processes with the Priority scheduling algorithm for given priorities in order (2,1,3,4).
5. Suppose that the following processes arrive for execution at the times indicated. Each process will run for the amount of time listed. In answering the questions, use non-preemptive scheduling, and base all decisions on the information you have at the time the decision must be made.

Process	Arrival Time	Burst Time
P1	0.0	8
P2	0.4	4
P3	1.0	1

Table 2: Process Scheduling Table

- What is the average turnaround time for these processes with the FCFS scheduling algorithm?
- What is the average turnaround time for these processes with the SJF scheduling algorithm?
- The SJF algorithm is supposed to improve performance, but notice that we chose to run process P1 at time 0 because we did not know that two shorter processes would arrive soon. Compute what the average turnaround time will be if the CPU is left idle for the first 1 unit and then SJF scheduling is used. Remember that processes P1 and P2 are waiting during this idle time, so their waiting time may increase. This algorithm could be called future-knowledge scheduling.

Learning Outcome: Students will be able to analyze process scheduling algorithms and synchronization mechanisms for efficient process coordination.

Web Resources:

- Computer Science and Engineering - Noc: Scheduling. [Online] <https://archive.nptel.ac.in/courses/106/105/106105214/>
- CPU Scheduling Algorithms - GeeksforGeeks: <https://www.geeksforgeeks.org/cpu-scheduling-in-operating-systems/>
- Process Synchronization - Programiz: <https://www.geeksforgeeks.org/introduction-of-process-synchronization/>

Unit III Deadlocks (7-8)

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

Description:

This unit covers deadlocks, a critical issue in operating systems where processes compete for resources indefinitely. It explores deadlock characterization, various handling methods, including prevention, avoidance, and detection, as well as strategies for recovery, ensuring efficient resource management and system stability.

Exercises:

- Consider the following snapshot of a system:

Process	Allocation				Max				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P0	0	0	1	2	0	0	1	2	1	5	2	0
P1	1	0	0	0	1	7	5	0				
P2	1	3	5	4	2	3	5	6				
P3	0	6	3	2	0	6	5	2				
P4	0	0	1	4	0	6	5	6				

Answer the following questions using the **Bankers Algorithm**:

- What is the content of the matrix **Need**?
- Is the system in a safe state?
- If a request from process P1 arrives for (0,4,2,0), can the request be granted immediately?

2. Consider the following snapshot of a system:

Process	Allocation				Max			
	A	B	C	D	A	B	C	D
P0	3	0	1	4	5	1	1	7
P1	2	2	1	0	3	2	1	1
P2	3	1	2	1	3	3	2	1
P3	0	5	1	0	4	6	1	2
P4	4	2	1	2	6	3	2	5

Using the **Bankers Algorithm**, determine whether or not each of the following states is unsafe. If the state is safe, illustrate the order in which the processes may complete. Otherwise, illustrate why the state is unsafe.

- Available = (0, 3, 0, 1)
- Available = (1, 0, 0, 2)

3. Consider the following snapshot of a system:

Process	Allocation				Max				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P0	2	0	0	1	4	2	1	2	3	3	2	1
P1	3	1	2	1	5	2	5	2				
P2	2	1	3	0	2	3	1	6				
P3	1	3	1	2	1	4	2	4				
P4	1	4	3	2	3	6	6	5				

Answer the following questions using the **Bankers Algorithm**:

- (a) Illustrate that the system is in a safe state by demonstrating an order in which the processes may complete.
- (b) If a request from process P1 arrives for (1, 1, 0, 0), can the request be granted immediately?
- (c) If a request from process P4 arrives for (0, 0, 2, 0), can the request be granted immediately?

Learning Outcome: Students will be able to understand deadlock concepts and apply strategies for prevention, avoidance, detection, and recovery.

Web Resources:

1. Computer Science and Engineering - Noc: Deadlock. [Online] <https://archive.nptel.ac.in/courses/106/105/106105214/>
2. Deadlock - GeeksforGeeks. [Online] <https://www.geeksforgeeks.org/introduction-of-deadlock-in-operating-system/>
3. Deadlock StudyTonight. [Online] <https://www.studytonight.com/operating-system/tests/20.php>

Unit IV Memory Management (9-11)

Memory Management Strategies:: Background, Swapping, Contiguous Memory Allocation, Segmentation, Paging.

Virtual Memory Management: Background, Demand Paging, Copy-on-Write, Page Replacement-FIFO, LRU, OPTIMAL, Thrashing.

Description:

This unit focuses on memory management strategies, including swapping, contiguous allocation, segmentation, and paging, to optimize system performance. It also covers virtual memory concepts such as demand paging, copy-on-write, and page replacement algorithms like FIFO, LRU, and OPTIMAL, addressing issues like thrashing for efficient memory utilization.

Exercises:

1. Given six memory partitions of 300 KB, 600 KB, 350 KB, 200 KB, 750 KB, and 125 KB (in order), how would the first-fit, best-fit, and worst-fit algorithms place processes of size 115 KB, 500 KB, 358 KB, 200 KB, and 375 KB (in order)? Rank the algorithms in terms of how efficiently they use memory.
2. Consider a logical address space of 256 pages with a 4-KB page size, mapped onto a physical memory of 64 frames.
 - (a) How many bits are required in the logical address?
 - (b) How many bits are required in the physical address?
3. Consider a computer system with a 32-bit logical address and 4-KB page size. The system supports up to 512 MB of physical memory. How many entries are there in each of the following?
4. Consider the following segment table:

Segment	Base	Length
0	219	600
1	2300	14
2	90	100
3	1327	580
4	1952	96

What are the physical addresses for the following logical addresses?

- (a) (0,430)
- (b) (1,10)
- (c) (2,500)
- (d) (3,400)
- (e) (4,112)

5. Consider the following page-replacement algorithms. Rank these algorithms on a five-point scale from bad to perfect according to their page-fault rate. Separate those algorithms that suffer from Beladys anomaly from those that do not.

- (a) LRU replacement
- (b) FIFO replacement
- (c) Optimal replacement

Learning Outcome: Students will be analyzing memory management strategies and virtual memory techniques for efficient resource allocation.

Web Resources:

1. **Computer Science and Engineering - Noc: Memory Management.** [Online] <https://archive.nptel.ac.in/courses/106/105/106105214/>
2. **Memory Management - GeeksforGeeks.** [Online] <https://www.geeksforgeeks.org/memory-management-in-operating-system/>
3. **Virtual Memory - GeeksforGeeks.** [Online] <https://www.geeksforgeeks.org/virtual-memory-in-operating-system/>

Unit V Storage Structure (12-14)

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

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

File System Implementation: Allocation Methods, Free-Space Management
Description:

This unit covers mass-storage structures and disk scheduling techniques to enhance data access efficiency. It explores RAID configurations for reliability and performance. Additionally, it delves into file system concepts, access methods, directory structures, and implementation techniques, including allocation methods and free-space management for effective storage utilization.

Exercises:

1. Suppose that a disk drive has 5,000 cylinders, numbered 0 to 4,999. The drive is currently serving a request at cylinder 2,150, and the previous request was at cylinder 1,805. The queue of pending requests, in FIFO order, is:

2, 069, 1, 212, 2, 296, 2, 800, 544, 1, 618, 356, 1, 523, 4, 965, 368

Starting from the current head position, what is the total distance (in cylinders) that the disk arm moves to satisfy all the pending requests for each of the following disk-scheduling algorithms?

- (a) FCFS
 - (b) SSTF
 - (c) SCAN
 - (d) LOOK
 - (e) C-SCAN
 - (f) C-LOOK
2. Write a program that implements the following disk-scheduling algorithms:
 - (a) FCFS
 - (b) SSTF
 - (c) SCAN
 - (d) C-SCAN
 - (e) LOOK
 - (f) C-LOOK

Your program will service a disk with 5,000 cylinders numbered 0 to 4,999. The program will generate a random series of 1,000 cylinder requests and service them according to each of the algorithms listed above. The program will be passed the initial position of the disk head (as a parameter on the command line) and report the total amount of head movement required by each algorithm.

3. Consider a file system that uses inodes to represent files. Disk blocks are 8 KB in size, and a pointer to a disk block requires 4 bytes. This file system has 12 direct disk blocks, as well as single, double, and triple indirect disk blocks. What is the maximum size of a file that can be stored in this file system?
4. Consider a file system on a disk that has both logical and physical block sizes of 512 bytes. Assume that the information about each file is already in memory. For each of the three allocation strategies (contiguous, linked, and indexed), answer these questions:
 - (a) How is the logical-to-physical address mapping accomplished in this system? (For the indexed allocation, assume that a file is always less than 512 blocks long.)
 - (b) If we are currently at logical block 10 (the last block accessed was block 10) and want to access logical block 4, how many physical blocks must be read from the disk?

Learning Outcome: Students will be understand mass-storage structures, disk scheduling, RAID, and file system implementation techniques.

Web Resources:

1. Computer Science and Engineering - Noc: File Systems and Secondary Storage. [Online] <https://archive.nptel.ac.in/courses/106/105/106105214/>
2. Mass-Storage Structure T Point Tech. [Online] <https://www.tpointtech.com/mass-storage-structure-in-operating-systems>
3. File Systems - GeeksforGeeks. [Online] <https://www.geeksforgeeks.org/file-systems-in-operating-system/>

Textbooks

1. Abraham Silberschatz, Peter B. Galvin, and Greg Gagne, *Operating System Concepts*, 9th ed., John Wiley & Sons (Asia) Pvt. Ltd, 2018.

Reference Books

1. Dhananjay M. Dhamdhare, *Operating Systems: A Concept-Based Approach*, 3rd ed., McGraw-Hill Education India Pvt. Ltd, 2017.
2. William Stallings, *Operating System: Internals and Design Principles*, 8th ed., Prentice Hall, 2014.
3. Andrew S. Tanenbaum, *Modern Operating Systems*, 4th ed., PHI, 2016.

Web Resources

1. **Operating System Fundamentals - NPTEL:** [Online] <https://archive.nptel.ac.in/courses/106/105/106105214/>
2. **Operating System - NPTEL:** [Online] <https://archive.nptel.ac.in/courses/106/102/106102132/>
3. **Introduction to Operating Systems - NPTEL:** [Online] <https://archive.nptel.ac.in/courses/106/106/106106144/>
4. **Coursera: Introduction to Operating System Specialization:** [Online] <https://www.coursera.org/specializations/codio-introduction-operating-systems>

24IT203 - COMPUTER ORGANIZATION & ARCHITECTURE			
Course Category:	Program Core	Credits:	3
Course Type:	Theory	Lecture-Tutorial-Practice:	3-0-0
Prerequisites:	-	Continuous Evaluation:	40
		Semester End Evaluation:	60
		Total Marks:	100

Course Description

This course provides the basic knowledge necessary to understand the hardware operation of digital computers. It introduces the fundamental concepts of computer organization and architecture, covering the fundamental principles behind the design and operation of modern computer systems. It explores hardware components, instruction sets, memory hierarchy, input/output (I/O) mechanisms, and memory organization.

Course Objectives

- Understand the basic structure and functioning of computer systems.
- Learn about instruction set architecture (ISA) and how the CPU executes instructions.
- Explore memory organization and cache mechanisms for efficient data access.
- Understand I/O systems, buses, and interrupt handling mechanisms.
- Analyze RISC vs. CISC architectures.

Course Outcomes

At the end of the course, the student will be able to:

- CO1:** Apply Boolean Algebra and Karnaugh Maps (K-maps) to simplify Boolean functions for Logic Circuit Design. [K3]
- CO2:** Design combinational & sequential circuits, Multiplexers, Decoders, Bus, Registers, and Counters.[K3]
- CO3:** Describe Hardwired and Microprogrammed control units of a digital computer system. [K2]
- CO4:** Implement arithmetic, logic, and shift micro-operations for a basic computer design.[K3]
- CO5:** Analyze various CPU organizations, instruction formats, addressing modes, RISC & CISC characteristics, data transfer mechanisms, memory, and I/O organization. [K4]

Mapping of Course Outcomes to Program Outcomes

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

Course Content

Unit I: Digital Logic Circuits

Topics: Logic Gates, Boolean Algebra, Map Simplification, Combinational Circuits, Flip-Flops, Sequential Circuits.

Digital Components: Decoders, Multiplexers, Registers, Shift Registers, Binary Counters.

Description: This unit provides a comprehensive introduction to digital logic circuits and digital components. It covers logic gates, Boolean algebra, combinational and sequential circuits, and advanced digital components such as decoders, multiplexers, registers, and counters. Students will gain theoretical knowledge and hands-on experience in designing and analyzing digital systems.

Learning Outcomes:

- Simplify logic expressions using Boolean rules and K-Maps.
- Design and implement combinational and sequential circuits.
- Analyze and construct flip-flop-based circuits like registers and counters.
- Apply decoders, multiplexers, and shift registers in digital systems.

Web Resources:

1. RealDigital.org: <https://realdigital.org> (Multiplexers and Decoders)
2. GeeksforGeeks: digital-electronics-and-logic-design-tutorials

Unit II: Register Transfer and Micro-Operations

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

Description: This unit explores the fundamental concepts of register transfer, micro-operations, and basic computer organization. It covers register transfer language, bus and memory transfers, arithmetic and logic micro-operations, shift operations, and the arithmetic logic shift unit.

Learning Outcomes:

- Understand data transfer mechanisms between registers & memory.
- Construct a common bus system with multiple registers.
- Define and implement basic arithmetic, logic, and shift micro-operations.

Web Resources:

1. Unacademy: <https://unacademy.com> (Register Transfer Language and Micro Operations)
2. GeeksforGeeks: [basic-computer-instructions](https://www.geeksforgeeks.org/basic-computer-instructions/)
3. NPTEL: <https://nptel.ac.in/courses/106102062>

Unit III: Microprogrammed Control and CPU Organization

Topics: Control Memory, Address Sequencing, Micro-Program Example, Design of Control Unit.

Central Processing Unit: General Register Organization, Stack Organization, Instruction Formats, Addressing Modes, RISC and CISC Characteristics.

Description: This unit focuses on the design of a microprogrammed control unit, where control signals are generated by a sequence of microinstructions stored in control memory. It also covers various CPU organizations and addressing modes.

Learning Outcomes:

- Design the microprogrammed control unit.
- Analyze the organization of registers, instruction formats, and addressing modes.

Web Resources:

1. Byjus: <https://byjus.com> (Microprogrammed Control Unit)
2. Slideserve: [central-processing-unit-powerpoint-ppt-presentation](https://www.slideserve.com/central-processing-unit-powerpoint-ppt-presentation)
3. NPTEL: <https://archive.nptel.ac.in/courses/106/105/106105163/>

Unit IV: Computer Arithmetic and Pipeline Processing

Topics: Addition and Subtraction, Multiplication Algorithms, Division Algorithms.

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

Description: This unit involves algorithms for performing basic arithmetic operations such as addition, subtraction, multiplication, and division, crucial for processing numerical data efficiently. It also focuses on Pipeline techniques used to enhance the execution speed of instructions.

Learning Outcomes:

- Implement various arithmetic operations.
- Perform stepwise execution of instructions through different pipeline stages.

Web Resources:

1. GeeksforGeeks: arithmetic-pipeline-and-instruction-pipeline
2. W3Schools: <https://www.w3schools.com> (Computer Arithmetic)
3. Slideshare.net: <https://www.slideshare.net> (Pipelining and Vector Processing)
4. IIT Delhi: <https://www.cse.iitd.ac.in/~srsarangi/archbook/archbook.pdf>

Unit V: Memory Organization and Input-Output Systems

Topics: Memory Hierarchy, Associative Memory, Cache Memory.

Input-Output Organization: Input-Output Interface, Asynchronous Data Transfer, Modes of Transfer, Priority Interrupt, Direct Memory Access (DMA).

Description: This unit covers the structure and management of different types of memory used in computer systems. It focuses on memory hierarchy, associative memory, and cache memory, which play a crucial role in optimizing data retrieval and storage. It also discusses efficient data transfer mechanisms such as interrupt-driven I/O and Direct Memory Access (DMA).

Learning Outcomes:

- Analyze the trade-offs between speed, cost, and capacity in the memory hierarchy.
- Understand the functioning of Associative memory and cache memory.
- Analyze Asynchronous Data Transfer Mechanisms and different modes of data transfer.

Web Resources:

1. GeeksforGeeks: io-interface-interrupt-dma-mode
2. Slideshare: unit-iv-memory-and-io-organization

Text Books

1. M. Morris Mano, *Computer System Architecture*, Revised Third Edition, Pearson Publications, 2020.

Reference Books

1. J.P. Hayes, *Computer Architecture and Organization*, TMH, International Second Revised Edition, 1998.
2. William Stallings, *Computer Organization and Architecture*, Ninth Edition, Pearson/PHI, 2013.
3. Andrew S. Tanenbaum, *Structured Computer Organization*, Fifth Edition, PHI/-Pearson, 2009.
4. Carl Hamacher, Zvonko Vranesic, Safwat Zaky, *Computer Organization*, Sixth Edition, McGraw Hill Publication, 2002.

E-Resources and Other Digital Material

1. Prof. D. Roychoudhury, IIT Kharagpur: Youtube
2. Prof. S. Raman, IIT Madras: <http://www.nptelvideos.com/course.php?id=396>
3. Prof. Kamakoti, IIT Chennai: <https://www.youtube.com/watch?v=MIWTxHbPBA0>
4. Prof. Anshul Kumar, IIT Delhi: computer-architecture-kumar-iit-delhi
5. Prof. P.K. Biswas, IIT Kharagpur: youtube

24IT204-OBJECT ORIENTED PROGRAMMING USING C++			
Course Category:	Program Core	Credits:	3
Course Type:	Practical	Lecture-Tutorial-Practice:	3-0-0
Prerequisites:	Problem Solving using python	Continuous Evaluation:	60
		Semester End Evaluation:	40
		Total Marks:	100

Course Description

This course provides an in-depth understanding of C++ programming with a focus on object-oriented concepts and their real-world applications. Students will explore fundamental programming constructs, class hierarchies, exception handling, and template-based programming. Emphasis is placed on writing efficient, reusable, and maintainable code while developing problem-solving skills essential for software development.

Course Objectives

- Develop a strong foundation in the core concepts of C++ programming and its key features.
- Understand and implement object-oriented principles to design efficient and reusable software.
- Utilize exception handling techniques to ensure robust and error-free program execution.
- Explore the use of templates to create generic and flexible code structures.

Course Outcomes

At the end of the course, the student will be able to:

- CO1:** Outline the essential features and elements of the C++ programming language. [K2]
- CO2:** Identify class hierarchies using the object-oriented design process. [K3]
- CO3:** Apply exception handling mechanism to handle errors occurring at runtime. [K3]
- CO4:** Apply Generic Programming Concepts for efficient and reusable code development.
- CO5:** Summarize generic classes with C++ templates. [K2]

Mapping of Course Outcomes to Program Outcomes

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

Course Content

Unit - I: Introduction to Object Oriented Programming (1-2)

Object Oriented Programming: Introduction, Encapsulation, Polymorphism, Inheritance, Dynamic binding, Structure of C++ program.

Classes & Objects: Classes, Structures vs Classes, Unions vs Classes, Friend Functions, Friend Classes, Inline functions, Constructors default, parameterized, Static Class Members Constructors and Destructors. The Scope Resolution Operator, Passing Objects to Functions, Returning Objects, this Pointer.

Description:

This unit introduces the fundamental concepts of object-oriented programming, including encapsulation, inheritance, polymorphism, and dynamic binding. It covers the structure of C++ programs, the role of classes and objects, constructors, destructors, and memory management.

Learning Outcome: Students will be able to understand and implement the fundamental concepts of object-oriented programming, design efficient classes and objects, and utilize various OOP features such as encapsulation, inheritance, and polymorphism in C++.

Web Resources:

1. NPTEL Object-Oriented Programming in C++ [Online] <https://nptel.ac.in/courses/106105151>
2. GeeksforGeeks C++ Object-Oriented Programming Basics [Online] <https://www.geeksforgeeks.org/object-oriented-programming-in-cpp/>
3. Tutorialspoint C++ Classes and Objects [Online] https://www.tutorialspoint.com/cplusplus/cpp_classes_objects.htm

Unit - II:

Overloading: Function Overloading, Overloading Constructor Functions, Copy Constructors, Operator Overloading, creating a Member Operator Function, Operator Overloading Using a Friend Function, overloading new and delete.

Inheritance: Base-Class Access Control, Inheritance and protected Members, Inheriting Multiple Base Classes, Constructors, Destructors and Inheritance, Virtual Base Classes.

Description:

This unit explores function overloading, operator overloading, and constructor overloading, which enhance code reusability and efficiency. It also covers inheritance, access control mechanisms, and multiple base class inheritance, enabling hierarchical class structures in object-oriented design.

Learning Outcome: Students will gain proficiency in function, operator, and constructor overloading for code reusability. They will learn inheritance mechanisms and access control for hierarchical class structures. Multiple inheritance and its practical applications will also be explored.

Web Resources:

1. GeeksforGeeks Function and Operator Overloading in C++ [Online] <https://www.geeksforgeeks.org/function-overloading-c/>
2. Tutorialspoint C++ Inheritance [Online] https://www.tutorialspoint.com/cplusplus/cpp_inheritance.htm
3. NPTEL Object-Oriented Programming: Inheritance and Overloading [Online] <https://nptel.ac.in/courses/106105151/module2>

Unit - III:

Virtual Functions: Calling a Virtual Function through a Base Class, Virtual attribute inheritance, Virtual functions are hierarchical, Pure Virtual Functions, early vs. Late Binding, Abstract Class.

Exception Handling: Exception Handling Fundamentals, catching class types, using multiple catch, Handling Derived Class Exceptions, Exception Handling Options Catching all exceptions, Restricting Exceptions, Rethrowing an Exception.

Description:

This unit covers dynamic polymorphism using virtual functions, ensuring flexibility in object-oriented programming. Exception handling techniques are introduced to manage runtime errors.

Learning Outcome: Students will implement virtual functions for dynamic polymorphism and exception handling for runtime errors. These skills will help them write efficient and flexible C++ programs.

Web Resources:

1. GeeksforGeeks Virtual Functions in C++ [Online] <https://www.geeksforgeeks.org/virtual-function-cpp/>
2. GeeksforGeeks Exception Handling in C++ [Online] <https://www.geeksforgeeks.org/exception-handling-c/>

Unit - IV:

Templates: Generic Functions, overloading a Generic Function, Overloading a function Template, Generic classes and case studies.

Description:

This unit introduces templates in C++, a powerful feature that enables generic programming by allowing functions and classes to work with multiple data types. Students will learn about generic functions, overloading generic functions, function template overloading, and generic classes, which help in writing reusable and efficient code. Templates reduce redundancy, improve maintainability, and provide type safety while optimizing performance. Mastering these concepts is essential for working with the Standard Template Library (STL) and developing scalable C++ applications.

Learning Outcome: After completing this unit, students will understand the fundamentals of generic programming in C++ and its role in enhancing code reusability and flexibility. They will learn to implement generic functions using function templates and apply function overloading to handle multiple data types efficiently. The unit also covers overloading function templates, enabling students to create more adaptable and specialized template functions. Additionally, students will explore generic classes, allowing them to design reusable and type-independent data structures. By mastering these concepts, students will be able to write modular, maintainable, and efficient C++ programs suitable for real-world applications.

Web Resources:

1. NPTEL C++ Templates and Exception Handling [Online] <https://nptel.ac.in/courses/106105151/module3>

Unit - V:

C++ Standard Template Library: Algorithms - Searching, Sorting.

Sequence Containers: Vectors, Strings, Lists, Dequeues.

Iterators: as Smart Pointers, as an Interface, matching algorithms with containers.

Description:

This unit introduces the Standard Template Library (STL), which provides a collection of generic classes and functions for efficient data management. It covers algorithms, sequence containers, and iterators to enhance C++ programming capabilities with reusable and optimized components.

Learning Outcome: Students will develop expertise in using the Standard Template Library (STL) for efficient data handling. They will apply STL containers, iterators, and algorithms to simplify operations. This knowledge will enhance their ability to write optimized and modular C++ code.

Web Resources:

1. GeeksforGeeks C++ Standard Template Library (STL) [Online] <https://www.geeksforgeeks.org/cpp-stl-tutorial/>
2. Tutorialspoint C++ STL Containers and Algorithms [Online] https://www.tutorialspoint.com/cplusplus/cpp_stl.htm
3. NPTEL Standard Template Library in C++ [Online] <https://nptel.ac.in/courses/106105151/module4>

Textbooks

1. Herbert Schildt, *C++: The Complete Reference*, Fourth Edition, The McGraw-Hill Companies, 2003.
2. Robert Lafore, *Object-Oriented Programming in C++*, Fourth Edition, Sams Publishing, USA, 2002.

Reference Books

1. Ulla Kirch-Prinz & Peter Prinz, *A Complete Guide to Programming in C++*, Jones and Bartlett Publishers, Canada.

Web Resources

1. **Dr. Partha Prathim Das, Professor, IIT Kharagpur, Programming in C++ (2016):** [Online] <https://nptel.ac.in/courses/106/105/106105151/>
2. **Dr. Abiram Ranade, Professor, IIT Bombay, Introduction to Programming through C++ (2016):** [Online] <https://nptel.ac.in/courses/106/101/106101208/>
3. **Jesse Dunietz, Instructional Designer, Massachusetts Institute of Technology, USA (2011):** [Online] [introduction-to-c-january-iap](#)
4. **GeeksforGeeks - A comprehensive material from a pool of developers:** [Online] <https://www.geeksforgeeks.org/c-plus-plus/>
5. **Anh Le, Programming in C++: A Hands-on Introduction Specialization:** [Online] <https://www.coursera.org/specializations/hands-on-cpp>

24IT281-OBJECT ORIENTED PROGRAMMING USING C++ LAB			
Course Category:	Programme Laboratory (PL)	Credits:	1.5
Course Type:	Practical	Lecture-Tutorial-Practice:	0-0-3
Prerequisites:	Problem Solving with C, Data Structures	Continuous Evaluation:	60
		Semester End Evaluation:	40
		Total Marks:	100

Course Description

This laboratory course provides hands-on experience in implementing object-oriented programming concepts using C++. Students will develop programs that incorporate classes, inheritance, polymorphism, exception handling, and templates. Through practical exercises, they will enhance their problem-solving skills, gain proficiency in debugging, and write efficient, reusable, and maintainable code. The course aims to bridge theoretical knowledge with practical application, preparing students for real-world software development challenges.

Course Objectives

1. Implement fundamental object-oriented programming (OOP) concepts such as classes and objects in C++.
2. Demonstrate the use of constructors, destructors, and operator overloading to manipulate class behavior effectively.
3. Apply inheritance and polymorphism to design modular and extensible applications.
4. Implement exception handling mechanisms to ensure robustness in C++ programs.
5. Utilize templates and the Standard Template Library (STL) to write reusable and efficient generic code.
6. Develop and test real-world applications using OOP principles, improving debugging and problem-solving skills.

Course Outcomes

At the end of the course, the student will be able to:

- CO1:** Implement object-oriented programming concepts such as classes, objects, and encapsulation in C++. [K3]
- CO2:** Develop programs using constructors, destructors, and operator overloading to enhance code reusability. [K3]

- CO3:** Demonstrate the application of inheritance and polymorphism in real-world scenarios. [K3]
- CO4:** Implement exception handling mechanisms to improve the reliability and stability of applications. [K3]
- CO5:** Design generic functions and classes using templates, improving code flexibility and reusability. [K3]
- CO6:** Build and test object-oriented applications to solve real-world problems effectively. [K3]

Mapping of Course Outcomes to Program Outcomes

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	3	2	3	2	2				1	2	3	3	2
CO2	3	2	3	2	2				1	2	3	3	2
CO3	3	3	3	2	2				1	2	3	3	3
CO4	3	3	3	2	2				1	2	3	3	3
CO5	3	3	3	3	2				1	3	3	3	3
CO6	3	3	3	3	2				1	3	3	3	3

Course Structure

- Classes and Objects** - Learn the fundamentals of classes and objects with single and multiple object implementations.
- Constructors & Inline Functions** - Explore constructors for object initialization and use inline functions for performance optimization.
- Static Data Members and Static Member Functions** - Understand the shared nature of static members and functions in C++.
- Passing Objects to Functions & Friend Functions** - Master passing objects to functions and implementing friend functions.
- Constructor Overloading** - Implement multiple constructors and understand the concept of copy constructors.
- Operator Overloading** - Overload binary, unary, and memory management operators for custom functionality.
- Inheritance** - Learn various types of inheritance and constructor invocation in C++.

8. **Virtual Functions and Abstract Classes** - Implement dynamic polymorphism using virtual functions and abstract classes.
9. **Exception Handling** - Handle runtime errors and create custom exceptions in C++.
10. **Generic Templates - Class Templates** - Implement function and class templates for generic programming.
11. **Standard Template Library (STL)** - Work with STL containers like vectors, lists, deques, and strings.
12. **Case Study Application Development** - Apply all concepts to real-world mini-projects like bank systems and ticket booking.

Text Books

1. Herbert Schildt, *C++: The Complete Reference*, Fourth Edition, The McGraw-Hill Companies, 2003.
2. Robert Lafore, *Object-Oriented Programming in C++*, Fourth Edition, Sams Publishing, USA, 2002.

Reference Books

1. Ulla Kirch-Prinz & Peter Prinz, *A Complete Guide to Programming in C++*, Jones and Bartlett Publishers, Canada.

Web Resources

1. **Dr. Partha Pratim Das**, Professor, IIT Kharagpur, Programming in C++, 2016, <https://nptel.ac.in/courses/106/105/106105151/>
2. **Dr. Abiram Ranade**, Professor, IIT Bombay, Introduction to programming through C++, 2016, <https://nptel.ac.in/courses/106/101/106101208/>
3. **Jesse Duniety**, Instructional designer, Massachusetts Institute of Technology, USA, 2011, introduction-to-c-january-iap-2011/index
4. A comprehensive material from a pool of developers at Geeks for Geeks webpage, <https://www.geeksforgeeks.org/c-plus-plus/>
5. **Anh Le**, Programming in C++: A Hands-on Introduction Specialization, <https://www.coursera.org/specializations/hands-on-cpp>

24IT282-ADVANCED DATA STRUCTURES AND ALGORITHMS LAB			
Course Category:	Laboratory	Credits:	1.5
Course Type:	Practical	Lecture-Tutorial-Practice:	0-0-3
Prerequisites:	Data Structures	Continuous Evaluation:	60
		Semester End Evaluation:	40
		Total Marks:	100

Course Description

The Advanced Data Structures and Algorithms Lab is designed to provide hands-on experience in implementing and experimenting with advanced data structures and algorithms. This lab course complements theoretical knowledge with practical application, focusing on the design, implementation, and optimization of sophisticated data structures and algorithms. Students can apply various algorithm techniques to solve engineering problems, combinatorial problems and heuristic approaches.

Course Objectives

1. Design and implement advanced data structures and algorithms.
2. Apply various algorithm techniques for providing solutions to engineering problems.
3. Design and implement efficient algorithms for a variety of computational problems.
4. Design algorithms for specified problems and analyze their time and space complexity.
5. Solve combinatorial problems using backtracking, branch and bound methods.
6. Implement and manage AVL trees, Red Black Trees, Heap trees, and B-trees, focusing on insertion, deletion, and balancing operations.

Course Outcomes

At the end of the course, the student will be able to:

- CO1:** Compute time complexity of a given algorithm. [K3]
- CO2:** Implement divide and conquer technique with time complexity analysis. [K3]
- CO3:** Design the algorithms for optimization problems using greedy and dynamic programming. [K4]
- CO4:** Apply backtracking technique for providing solutions to combinatorial problems. [K3]
- CO5:** Perform operations on balanced data structures - AVL and Heaps. [K3]

CO6: Solve scenario-based problems using appropriate data structures and design techniques. [K3]

Mapping of Course Outcomes to Program Outcomes

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	2	2	1									3	
CO2	2	2	1									1	2
CO3	2	1	1									2	2
CO4	2	2	2									3	1
CO5	2	1	1									1	2
CO6	2	3	3									2	2

Course Structure

Week 1: Programs on Time Complexity Analysis Techniques

1. Design an algorithm for the given problem and analyze the time complexity of the algorithm.
2. Design an algorithm for the given problem and analyze the time complexity of the algorithm.

Week 2 & 3: Programs on Divide and Conquer Technique

1. Sorting techniques: Merge Sort and Quick Sort
2. Find the minimum and maximum in a given array of elements
3. Design experiment using Divide and Conquer Technique

Week 4 & 5: Programs on Greedy Method

1. Implement fractional Knapsack problem
2. Find minimum cost spanning tree using Prims and Kruskals Algorithms
3. Implement Single Source Shortest Path problem
4. Implement job sequencing with deadlines problem

Week 6: Programs on Dynamic Programming

1. Implementation of All Pairs Shortest Path problem
2. Implementation of 0/1 Knapsack Problem

Week 7 & 8: Programs on Backtracking

1. Implement N-Queens Problem
2. Implement sum of subsets problem
3. Implement Graph coloring problem
4. Implement Hamiltonian cycle problem

Week 9: AVL Tree and Applications

1. Implementation of AVL tree operations
2. Design experiment using AVL-Tree

Week 10: Heap and Applications

1. Insert and delete operations on Min or Max Heap
2. Design experiment using Heap Tree

Week 11: Design Experiments/Scenario-Based Problem Solving Using Advanced Data Structures

Week 12: Design Experiments/Scenario-Based Problem Solving Using Algorithm Design Strategies

Text Books

1. Horowitz Sahni and Anderson-Freed, *Fundamentals of Data Structures in C*, 2nd edition, Universities Press, 2011.
2. Mark Allen Weiss, *Data Structure and Algorithm Analysis in C*, 2nd edition, Addison Wesley Publication, 2010.

Reference Books

1. Yedidyah Langsam, Moshe J. Augenstein, and Aaron M. Tenenbaum, *Data Structures using C and C++*, 2nd edition, Pearson Education, 1999.
2. R. F. Gilberg and B. A. Forouzan, *Data Structures: A Pseudocode Approach with C*, 2nd edition, Cengage Learning.

Web Resources

1. Erik Demaine, Advanced Data Structures, [MIT- OpenCourseWare]. (26, May, 2021). Available: <http://ocw.mit.edu/>
2. Dr. Naveen Garg, Department of Computer Science & Engineering, IIT Delhi, Lecture Series on Data Structures and Algorithms [NPTEL], (26, May, 2021) Available: <https://nptel.ac.in/courses/106/102/106102064/>
3. Data Structures and applications on, [GeeksforGeeks], (25, May, 2021) Available: <https://www.geeksforgeeks.org/data-structures/>
4. Data Structures and challenges [HackerRank], (25, May, 2021) Available: <https://www.hackerrank.com/domains/data-structures>

24IT283-WEB PROGRAMMING			
Course Category:	Programee Laboratory (PL)	Credits:	1
Course Type:	Practical	Lecture - Tutorial - Practice:	0-0-2
Prerequisites:	-	Continuous Evaluation:	40
		Semester End Evaluation:	60
		Total Marks:	100

Course Description

The Web Programming Lab provides hands-on training in designing and developing interactive web applications using HTML, CSS, and JavaScript. This course covers fundamental to advanced web technologies, including responsive design, client-side scripting, form handling, and validations. Students will gain practical experience in creating dynamic, user-friendly websites while working on real-world case studies like an Online Bookstore and a Student Course Hub. By the end of the course, students will be able to design, develop, and deploy fully functional web applications.

Course Objectives

1. Design and build both static and dynamic web pages using HTML, CSS, and JavaScript to create engaging and functional websites.
2. Utilize CSS for styling and layout, ensuring responsive and visually appealing web designs across different devices.
3. Implement JavaScript for client-side scripting to add interactivity, enhance user experience, and manage dynamic content updates.
4. Integrate HTML forms with JavaScript to validate user input, process form data efficiently, and improve usability.
5. Manipulate the DOM dynamically using JavaScript to update webpage content, handle events, and enhance user interactions.
6. Develop responsive and structured web applications by integrating Bootstrap, media queries, and layout techniques for optimal user experience across various screen sizes.

Course Outcomes

At the end of the course, the student will be able to:

CO1: Develop static and structured web pages using HTML and CSS. [K3]

CO2: Design and implement responsive web layouts using CSS techniques and Bootstrap. [K3]

CO3: Implement interactive features using JavaScript for client-side scripting. [K3]

CO4: Update DOM dynamically using JavaScript for real-time content. [K3]

CO5: Validate and process form data using JavaScript for improved user input. [K4]

CO6: Develop and deploy web pages with optimized performance and cross-browser compatibility. [K3]

Mapping of Course Outcomes to Program Outcomes

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	1										1		
CO2					3								3
CO3					2								2
CO4	1				3								1
CO5	1				2							2	
CO6	1				3							2	

Course Structure

1. Week 1: Introduction to HTML5 Structuring a Web Page

- Overview of HTML5 and the basic document structure.
- Using semantic elements: <header>, <footer>, <nav>, etc.
- Formatting content: <p>, , <i>, <mark>, etc.
- Adding images with and hyperlinks with <a>.

Case Studies:

- Online Bookstore: Create homepage structure.
- Personal Portfolio Website: Develop homepage layout.

2. Week 2: Lists, Tables, Frames, and Forms in HTML5

- Creating ordered, unordered, and description lists.
- Structuring tables using <table>, <tr>, <td>, etc.
- Embedding content using <iframe>.

- Handling user input with forms and new input types.

Case Studies:

- Online Bookstore: Contact form and book reviews via `<iframe>`.
- Personal Portfolio Website: Contact form with Google Map.

3. Week 3: Introduction to CSS3 Styling Web Pages

- Understanding inline, internal, and external CSS.
- Applying CSS selectors and styling text and fonts.
- Setting backgrounds and borders.

Case Studies:

- Online Bookstore: Apply custom fonts and colors.
- Personal Portfolio Website: Style profile section.

4. Week 4: CSS3 Box Model, Positioning, and Pseudo-Classes

- Understanding the CSS Box Model.
- Positioning techniques: static, relative, absolute, fixed.
- Working with pseudo-classes and pseudo-elements.

Case Studies:

- Online Bookstore: Implement structured book display.
- Personal Portfolio Website: Style navigation bar.

5. Week 5: CSS3 Flexbox, Grid Layouts, and CSS Variables

- Implementing Flexbox and CSS Grid.
- Creating responsive layouts.
- Using CSS variables and custom properties.

Case Studies:

- Online Bookstore: Organize book listings with CSS Grid.
- Personal Portfolio Website: Arrange sections using Flexbox.

6. Week 6: Responsive Web Design with Media Queries

- Implementing @media queries for responsiveness.
- Using viewport units and responsive typography.

Case Studies:

- Online Bookstore: Ensure mobile adaptability.
- Personal Portfolio Website: Create a mobile-friendly portfolio.

7. Week 7: Introduction to JavaScript Basics

- JavaScript syntax and variables.
- Data types, operators, and expressions.
- Handling user events with `addEventListener()`.

Case Studies:

- Online Bookstore: Interactive book search.
- Personal Portfolio Website: Theme toggle button.

8. Week 8: JavaScript Functions, Loops, and DOM Manipulation

- Writing functions and using loops.
- Modifying HTML dynamically using JavaScript.

Case Studies:

- Online Bookstore: Implement book filter.
- Personal Portfolio Website: Skills progress bar.

9. Week 9: JavaScript Validations and Form Handling

- Validating forms using HTML5 and JavaScript.
- Using regular expressions for validation.

Case Studies:

- Online Bookstore: Validate contact form input.
- Personal Portfolio Website: Ensure correct input handling.

10. Week 10: JavaScript API Fetching and Local Storage

- Fetching data using Fetch API and `async/await`.
- Using `localStorage` and `sessionStorage`.

Case Studies:

- Online Bookstore: Shopping cart with local storage.
- Personal Portfolio Website: Save user theme preferences.

11. Week 11: CSS3 Animations and Transitions

- Creating animations using `@keyframes`.
- Applying smooth CSS transitions.
- Using 3D transformations.

Case Studies:

- Online Bookstore: Animate book covers on hover.

- Personal Portfolio Website: Add smooth animations.

12. Week 12: Web Optimization, SEO, and Deployment

- Optimizing page speed and implementing SEO best practices.
- Deploying websites using GitHub Pages, Netlify, Firebase.

Final Projects:

- Online Bookstore: Fully functional e-commerce website.
- Personal Portfolio Website: Build and deploy live portfolio site.

Textbooks

1. Ben Frain, *Responsive Web Design with HTML5 and CSS*, 4th Edition, Packt Publishing, 2022.
2. David Flanagan, *JavaScript: The Definitive Guide*, 7th Edition, O'Reilly Media, 2020.

Reference Books

1. Dino Esposito, *Modern Web Development: Understanding Domains, Technologies, and User Experience*, 1st Edition, Microsoft Press, 2023.
2. Eric A. Meyer, Estelle Weyl, *CSS: The Definitive Guide*, 5th Edition, O'Reilly Media, 2023.
3. Julie C. Meloni, J Kymin, *Sams Teach Yourself HTML, CSS & JavaScript All in One*, 3rd Edition, Pearson, 2018.

Web Resources

1. The Modern JavaScript Tutorial. Available: <https://javascript.info/>
2. MDN Web Docs (Mozilla Developer Network). Available: <https://developer.mozilla.org/en-US/>
3. W3Schools Web Development Portal. Available: <https://www.w3schools.com/>
4. Udemy: Learn By Example The Foundations of HTML, CSS & JavaScript. Available: <https://www.udemy.com/course/learn-by-example-html-css-javascript>
5. CSS Tricks. Available: <https://css-tricks.com/>

24UC201-Universal Human Values-II			
Course Category:	MC	Credits:	3
Course Type:	Theory	Lecture-Tutorial-Practice:	3-0-0
Prerequisites:	UHV-I (in Student Induction Program)	Continuous Evaluation:	60
		Semester End Evaluation:	40
		Total Marks:	100

Course Description

The course is mandated by AICTE for B.Tech. students of all branches, preferably in the second year. It is intended to facilitate the development of a holistic and humane world vision. This course employs an innovative and effective methodology of self-exploration, self-verification on one's own right. It is presented as a systematic set of universal, rational and verifiable proposals about human reality, about the inherent harmony in the human being, the family, society, the entire nature and existence. It draws out the value or role of human being in the harmony at all these levels, which is essentially the scope of ethical human conduct. It further helps to find resolution of present-day challenges. The issues in professional ethics are analysed in the context of the understanding of harmony. While handling the course, the teacher is also a co-explorer along with the students.

Course Objectives

1. Help the students appreciate the essential complementarity between values and skills to ensure sustained happiness and prosperity which are the core aspirations of all human beings.
2. Facilitate the development of a holistic perspective among learners towards life and profession.
3. Facilitate the right understanding of happiness and prosperity based on correct understanding of human reality and the rest of existence..
4. Identify that a holistic perspective forms the basis of universal human values and movement towards value-based living in a natural way.
5. Solve combinatorial problems using backtracking, branch and bound methods.
6. Highlight plausible implications of holistic understanding in terms of ethical human conduct, trustful and mutually fulfilling human behavior and mutually enriching interaction with Nature.

Course Outcomes

At the end of the course, the student will be able to:

- CO1:** Apply the right understanding of the concepts of value education and basic human aspirations through self-exploration for the fulfillment of human aspirations [K3]
- CO2:** Analyse various aspects of the human being as the combination of Self and Body for attaining harmony at the level of human being (individual) [K4]
- CO3:** Apply the knowledge of nine universal values in human-human relationship for harmony at the level of family, and appreciate all the essential factors that help in attaining harmony at the level of society [K3]
- CO4:** Differentiate the characteristics and activities of various orders of Nature and study the mutual fulfillment among them, and also identify the existence as co-existence at all levels [K4]
- CO5:** Present sustainable solutions to various challenges in society and Nature, and identify that the solutions are practicable [K3]

Mapping of Course Outcomes to Program Outcomes

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1						H							H
CO2						H							H
CO3						H			M	M			H
CO4						H	H		M	M			H
CO5						H	H	H	H	H			H

Course Structure

Unit 1: Course introduction, need, basic guidelines, content and process for value education (6 lectures + 3 tutorials)

Contents: Purpose and motivation for the course, recapitulation from UHV-I, Self-exploration: what is it? its content and process, Natural acceptance and experimental validation as the process for self-exploration, Continuous happiness and prosperity a look at basic human aspirations. Right understanding, relationship and physical facility the basic requirements for fulfillment of aspirations of every human being with their correct priority, understanding happiness and prosperity correctly a critical appraisal of the current scenario, method to fulfill the above human aspirations: understanding and living in harmony at various levels.

Description: This unit deals with the correct meaning of right understanding, relationship and physical facility, value education, self-exploration, and introduces the two basic aspirations of human being and methods to fulfill them.

Examples/Applications/Case Studies:

- Examples to differentiate the meaning of happiness and prosperity in the current scenario and the meaning proposed in the course.
- Case studies related to the basic aspirations assumed by the students and the real basic aspirations.

Exercises/Projects/Practices: Practice sessions are to be included to discuss natural acceptance in human being as the innate acceptance for living with responsibility (living in relationship, harmony and co-existence) rather than as arbitrariness in choice based on liking-disliking.

Learning Outcomes:

- Explore students themselves, and start appreciating the need and relevance for the course.
- Identify that lack of understanding of human values is the root cause of most of the present-day challenges.
- Understand that verification on the basis of natural acceptance and experiential validation through living is the only way to verify right and wrong.

Unit 2: Understanding Harmony in the Human Being Harmony in Myself
(6 lectures + 3 tutorials)

Contents: Understanding human being as a co-existence of the sentient I and the material Body. Understanding the needs of Self (I) and Body happiness and physical facility, Understanding the Body as an instrument of I (I being the doer, seer and enjoyer). Understanding the characteristics and activities of I and harmony in I. Understanding the harmony of I with the Body: Sanyam and Health; correct appraisal of Physical needs, meaning of Prosperity in detail, Programs to ensure Sanyam and Health.

Description: This unit explores human being as a combination of two entities namely Self and Body. It proposes that the needs and functions of both Self and Body are different, and body as an instrument of the Self. It provides the aspects related to harmony within.

Examples/Applications/Case Studies:

- Examples to differentiate the functions and needs of both Self and Body.
- Case studies related to the Sanyam and Health and correct appraisal of physical needs that provides clarity on the real meaning of prosperity.

Exercises/Projects/Practices: Practice sessions are to be included to discuss the role others have played in making material goods available to me. Identifying from ones own life. Differentiate between prosperity and accumulation. Discuss program for ensuring health vs. dealing with disease.

Learning Outcomes:

- Distinguish the desires, needs and functions of the body and self.
- Aware of the activities of the Self and the sources of the desires of the Self.
- Apply the daily routine for maintaining the health of the body.

Unit 3: Understanding Harmony in the Family and Society Harmony in Human-Human Relationship (6 lectures + 3 tutorials)

Contents: Understanding values in human-human relationship; meaning of Justice (nine universal values in relationships) and program for its fulfillment to ensure mutual happiness; Trust and Respect as the foundational values of relationship. Understanding the meaning of Trust; Difference between intention and competence, Understanding the meaning of Respect, Difference between respect and differentiation; the other salient values in relationship. Understanding the harmony in the society (society being an extension of family); Resolution, Prosperity, Fearlessness (trust) and Co-existence as comprehensive Human Goals. Visualizing a universal harmonious order in society Undivided Society, Universal Order from family to world family.

Description: This unit explores the values in human-human relationships, which are essential for maintaining harmony in the family. It also explores the aspects necessary for bringing harmony in society.

Examples/Applications/Case Studies:

- Examples for exploring the nine values in human-human relationships in family.
- Case studies for visualizing a universal harmonious order in society and universal order.

Exercises/Projects/Practices: Practice sessions are to be included to reflect on relationships in family, hostel, and institute as an extended family. Real-life examples of teacher-student relationships, goal of education, etc. Gratitude as a universal value in relationships discuss with scenarios and elicit examples from students lives.

Learning Outcomes:

- Identify the correct meaning of respect, trust, and other values in human-human relationships.
- Understand the real cause of various societal challenges and solutions for addressing them.
- Develop an outline of a holistic model for social science and compare it with the existing model.

Unit 4: Understanding Harmony in Nature & Existence Whole Existence as Coexistence (6 lectures + 3 tutorials)

Contents: Understanding harmony in Nature, Inter-connectedness, and mutual fulfillment among the four orders of Nature recyclability and self-regulation in nature. Understanding Existence as Co-existence of mutually interacting units in all-pervasive space. Holistic perception of harmony at all levels of existence.

Description: This unit explores the four orders existing in Nature and their inter-connectedness. It clarifies that existence is co-existence of mutually interacting units. It proposes a desirable holistic perception of harmony at all levels of existence.

Examples/Applications/Case Studies:

- Examples to distinguish various units in all the four orders of Nature.
- Case studies related to recyclability and self-regulation in Nature.

Exercises/Projects/Practices: Practice sessions are to be included to discuss human beings as the cause of imbalance in nature (film Home can be used), pollution, depletion of resources, and role of technology, etc.

Learning Outcomes:

- Differentiate between the characteristics and activities of different orders and study the mutual fulfillment among them.
- Identify the need to take appropriate steps to ensure the right participation in Nature.
- Realize the mutually fulfilling mechanisms and interconnectedness existing in Nature.

Unit 5: Implications of the Above Holistic Understanding of Harmony on Professional Ethics

Contents: Natural acceptance of human values, Definitiveness of ethical human conduct, Basis for humanistic education, humanistic constitution, and humanistic universal order. Competence in professional ethics:

- Ability to utilize professional competence for augmenting universal human order.
- Ability to identify the scope and characteristics of people-friendly and eco-friendly production systems.
- Ability to identify and develop appropriate technologies and management patterns for the above production systems.

Case studies of typical holistic technologies, management models, and production systems. Strategy for transition from the present state to Universal Human Order:

- At the level of the individual: as socially and ecologically responsible engineers, technologists, and managers.
- At the level of society: as mutually enriching institutions and organizations.

Description: This unit deals with the definitiveness of ethical human conduct and the basis for humanistic education and competence in professional ethics. It provides case studies of typical holistic technologies, management models, and production systems.

Examples/Applications/Case Studies:

- Examples for holistic technologies, management models, and production systems.
- Case studies related to production systems, management models, and technologies.

Exercises/Projects/Practices: Practice exercises and case studies are to be taken up in practice (tutorial) sessions, e.g., to discuss ethical conduct as an engineer or scientist.

Learning Outcomes:

- Present sustainable solutions to the problems in society and nature and ensure those solutions are practicable and draw roadmaps to achieve them.
- Appreciate the right utilization of their knowledge in their field of Engineering.

Text Books

1. Gaur, R.R., Sangal, R., & Bagaria, G.P. (2010). *A Foundation Course in Human Values and Professional Ethics*. Excel Books Private Limited, New Delhi.
2. Gaur, R.R., Asthana, R., & Bagaria, G.P. (2019). *A Foundation Course in Human Values and Professional Ethics* (2nd revised edition). Excel Books Private Limited, New Delhi.

Reference Books

1. Jeevan Vidya: Ek Parichaya, A. Nagaraj, Jeevan Vidya Prakashan, Amarkantak (1999).
2. Human Values, A. N. Tripathi, New Age International Publishers, New Delhi (2004).
3. The Story of Stuff: The impact of overconsumption on the planet, our communities, and our health and how we can make it better, Annie Leonard, Free Press, New York (2010).
4. The Story of My Experiments with Truth: Mahatma Gandhi Autobiography, Mohandas Karamchand Gandhi, B. N. Publishing (2008).
5. Small is Beautiful: A Study of Economics as if People Mattered, E. F. Schumacher, Vintage Books, London (1993).
6. Slow is Beautiful: New Visions of Community, Cecile Andrews, New Society Publishers, Canada (2006).
7. Economy of Permanence, J. C. Kumarappa, Sarva-Seva-Sangh Prakashan, Varanasi (2017).
8. Bharat Mein Angreji Raj, Pandit Sunderlal, Prabhath Prakashan, Delhi (2018).
9. Rediscovering India, Dharampal, Society for Integrated Development of Himalayas (2003).
10. Hind Swaraj or Indian Home Rule, M. K. Gandhi, Navajivan Publishing House, Ahmedabad (1909).
11. India Wins Freedom: The Complete Version, Maulana Abul Kalam Azad, Orient Blackswan (1988).
12. The Life of Vivekananda and the Universal Gospel, Romain Rolland, Advaita Ashrama, India (2010).
13. Mahatma Gandhi: The Man Who Became One with the Universal Being, Romain Rolland, Srishti Publishers & Distributors, New Delhi (2002).

Web Resources

- **Textbook-1:** a-foundation-course-in-human-values-and-professional-ethics
- **AICTE SIP Youtube Channel:** youtube
- **AICTE UHV Teaching Learning Material:** <https://fdp-si.aicte-india.org/download.php#1>