III Semester

DATA STRUCTURES AND APPLICATIONS

Course Code:	21CS32	CIE Marks	50
Teaching Hours/Week (L:T:P: S)	3:0:2:0	SEE Marks	50
Total Hours of Pedagogy	40 T + 20 P	Total Marks	100
Credits	04	Exam Hours	03

Course Objectives:

CLO 1. Explain the fundamentals of data structures and their applications essential for implementing solutions to problems.
CLO 2. Illustrate representation of data structures: Stack, Queues, Linked Lists, Trees and Graphs.
CLO 3. Design and Develop Solutions to problems using Arrays, Structures, Stack, Queues, Linked Lists.
CLO 4. Explore usage of Trees and Graph for application development.
CLO 5. Apply the Hashing techniques in mapping key value pairs.

Teaching-Learning Process (General Instructions)

These are sample Strategies, which teachers can use to accelerate the attainment of the various course outcomes.

1. Lecturer method (L) need not to be only traditional lecture method, but alternative effective teaching methods could be adopted to attain the outcomes.
2. Use of Video/Animation to explain functioning of various concepts.
3. Encourage collaborative (Group Learning) Learning in the class.
4. Ask at least three HOT (Higher order Thinking) questions in the class, which promotes critical thinking.
5. Adopt Problem Based Learning (PBL), which fosters students' Analytical skills, develop design thinking skills such as the ability to design, evaluate, generalize, and analyze information rather than simply recall it.
6. Introduce Topics in manifold representations.
7. Show the different ways to solve the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding.

Module-1

Introduction: Data Structures, Classifications (Primitive & Non-Primitive), Data structure operations (Traversing, inserting, deleting, searching, and sorting). Review of Arrays. Structures: Array of structures Self-Referential Structures.
Dynamic Memory Allocation Functions. Representation of Linear Arrays in Memory, dynamically allocated arrays and Multidimensional Arrays.
Demonstration of representation of Polynomials and Sparse Matrices with arrays.

Textbook 1: Chapter 1: 1.2, Chapter 2: 2.2 - 2.7, Text Textbook 2: Chapter 1: 1.1 - 1.4, Chapter 3: 3.1 - 3.3, 3.5, 3.7, Chapter 4: 4.1 - 4.9, 4.14 Textbook 3: Chapter 1: 1.3

Laboratory Component:

1. Design, Develop and Implement a menu driven Program in C for the following Array Operations
   a. Creating an Array of N Integer Elements
   b. Display of Array Elements with Suitable Headings
   c. Exit.

   Support the program with functions for each of the above operations.

2. Design, Develop and Implement a menu driven Program in C for the following Array operations
   a. Inserting an Element (ELEM) at a given valid Position (POS)
   b. Deleting an Element at a given valid Position POS)

a. Create a DLL stack of N Professor's Data.

b. Create a DLL queue of N Professor's Data

Display the status of DLL and count the number of nodes in it.

Teaching-Learning Process	MOOC, Active Learning, Problem solving based on linked lists. https://nptel.ac.in/courses/106/102/106102064/ https://ds1-iiith.vlabs.ac.in/exp/linked-list/basics/overview.html https://ds1-iiith.vlabs.ac.in/List%20of%20experiments.html https://ds1-iiith.vlabs.ac.in/exp/linked-list/basics/overview.html https://ds1-iiith.vlabs.ac.in/List%20of%20experiments.html

Module-4

Trees 1: Terminologies, Binary Trees, Properties of Binary trees, Array and linked Representation of Binary Trees, Binary Tree Traversals - Inorder, postorder, preorder; Threaded binary trees, Binary Search Trees – Definition, Insertion, Deletion, Traversal, and Searching operation on Binary search tree. Application of Trees-Evaluation of Expression.

Textbook 1: Chapter 5: 5.1 –5.5, 5.7; Textbook 2: Chapter 7: 7.1 – 7.9

Laboratory Component:

1. Given an array of elements, construct a complete binary tree from this array in level order fashion. That is, elements from left in the array will be filled in the tree level wise starting from level 0. Ex: Input :
   arr[] = {1, 2, 3, 4, 5, 6}
   Output : Root of the following tree

```
     1
    / \
   2   3
  / \  /\
 4   5 6
```

2. Design, Develop and Implement a menu driven Program in C for the following operations on Binary Search Tree (BST) of Integers
   a. Create a BST of N Integers
   b. Traverse the BST in Inorder, Preorder and Post Order

Teaching-Learning Process	Problem based learning http://www.nptelvideos.in/2012/11/data-structures-and-algorithms.html https://ds1-iiith.vlabs.ac.in/exp/tree-traversal/index.html https://ds1-iiith.vlabs.ac.in/exp/tree-traversal/depth-first-traversal/dft-practice.html

Module-5

Trees 2: AVL tree, Red-black tree, Splay tree, B-tree.

Graphs: Definitions, Terminologies, Matrix and Adjacency List Representation of Graphs, Traversal methods: Breadth First Search and Depth FirstSearch.

Hashing: Hash Table organizations, Hashing Functions, Static and Dynamic Hashing.

Textbook 1: Chapter 10:10.2, 10.3, 10.4, Textbook 2:7.10 – 7.12, 7.15 Chapter 11: 11.2, Textbook 1: Chapter 6 : 6.1–6.2, Chapter 8 : 8.1-8.3, Textbook 2: 8.1 – 8.3, 8.5, 8.7

Textbook 3: Chapter 15:15.1, 15.2,15.3, 15.4,15.5 and 15.7

Laboratory Component:

1. Design, Develop and implement a program in C for the following operations on Graph (G) of cities
   a. Create a Graph of N cities using Adjacency Matrix.
   b. Print all the nodes reachable from a given starting node in a diagraph using DFS/BFS method.

2. Design and develop a program in C that uses Hash Function H:K->L as H(K)=K mod m(reminder method) and implement hashing technique to map a given key K to the address space L. Resolve the collision (if any) using linear probing.

Teaching-Learning Process	NPTL, MOOC etc. courses on trees and graphs. http://www.nptelvideos.in/2012/11/data-structures-and-algorithms.html

Course Outcomes (Course Skill Set)

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

CO 1. Identify different data structures and their applications.
CO 2. Apply stack and queues in solving problems.
CO 3. Demonstrate applications of linked list.
CO 4. Explore the applications of trees and graphs to model and solve the real-world problem.
CO 5. Make use of Hashing techniques and resolve collisions during mapping of key value pairs

Assessment Details (both CIE and SEE)

The weightage of Continuous Internal Evaluation (CIE) is 50% and for Semester End Exam (SEE) is 50%. The minimum passing mark for the CIE is 40% of the maximum marks (20 marks). A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to each subject/ course if the student secures not less than 35% (18 Marks out of 50) in the semester-end examination (SEE), and a minimum of 40% (40 marks out of 100) in the sum total of the CIE (Continuous Internal Evaluation) and SEE (Semester End Examination) taken together

Continuous Internal Evaluation:

Three Unit Tests each of **20 Marks (duration 01 hour)**

1. First test at the end of 5th week of the semester
2. Second test at the end of the 10th week of the semester
3. Third test at the end of the 15th week of the semester

Two assignments each of **10 Marks**

4. First assignment at the end of 4th week of the semester
5. Second assignment at the end of 9th week of the semester

Practical Sessions need to be assessed by appropriate rubrics and viva-voce method. This will contribute to **20 marks.**

- Rubrics for each Experiment taken average for all Lab components – 15 Marks.
- Viva-Voce– 5 Marks (more emphasized on demonstration topics)

The sum of three tests, two assignments, and practical sessions will be out of 100 marks and will be **scaled down to 50 marks**

(to have a less stressed CIE, the portion of the syllabus should not be common /repeated for any of the methods of the CIE. Each method of CIE should have a different syllabus portion of the course).

CIE methods /question paper has to be designed to attain the different levels of Bloom's taxonomy as per the outcome defined for the course.

Semester End Examination:

Theory SEE will be conducted by University as per the scheduled timetable, with common question papers for the subject (**duration 03 hours)**

1. The question paper will have ten questions. Each question is set for 20 marks. Marks scored shall be proportionally reduced to 50 Marks
2. There will be 2 questions from each module. Each of the two questions under a module (with a maximum of 3 sub-questions), **should have a mix of topics** under that module.

The students have to answer 5 full questions, selecting one full question from each module

Suggested Learning Resources:

Textbooks:

1. Ellis Horowitz and Sartaj Sahni, Fundamentals of Data Structures in C, 2nd Ed, Universities Press, 2014.
2. Seymour Lipschutz, Data Structures Schaum's Outlines, Revised 1st Ed, McGraw Hill, 2014.
3. Reema Thareja, Data Structures using C, 3rd Ed, Oxford press, 2012.

Reference Books:

1. Gilberg and Forouzan, Data Structures: A Pseudo-code approach with C, 2nd Ed, Cengage Learning,2014.
2. Jean-Paul Tremblay & Paul G. Sorenson, An Introduction to Data Structures with Applications,2nd Ed, McGraw Hill, 2013
3. A M Tenenbaum, Data Structures using C, PHI, 1989
4. Robert Kruse, Data Structures and Program Design in C, 2nd Ed, PHI, 1996.

Weblinks and Video Lectures (e-Resources):

1. http://elearning.vtu.ac.in/econtent/courses/video/CSE/06CS35.html
2. https://nptel.ac.in/courses/106/105/106105171/
3. http://www.nptelvideos.in/2012/11/data-structures-and-algorithms.html

Activity Based Learning (Suggested Activities in Class)/ Practical Based learning

- Real world problem solving using group discussion.
- Back/Forward stacks on browsers.
- Undo/Redo stacks in Excel or Word.
- Linked list representation of real-world queues -Music player, image viewer

Karnaugh maps: minimum forms of switching functions, two and three variable Karnaugh maps, four variable Karnaugh maps, determination of minimum expressions using essential prime implicants, Quine-McClusky Method: determination of prime implicants, the prime implicant chart, Petricks method, simplification of incompletely specified functions, simplification using map-entered variables

Textbook 1: Part B: Chapter 5 (Sections 5.1 to 5.4) Chapter 6 (Sections 6.1 to 6.5)

Laboratory Component:

1. Given a 4-variable logic expression, simplify it using appropriate technique and inplement the same using basic gates.

Teaching-Learning Process	1. Chalk and Board for numerical 2. Laboratory Demonstration

Module-3

Combinational circuit design and simulation using gates: Review of Combinational circuit design, design of circuits with limited Gate Fan-in, Gate delays and Timing diagrams, Hazards in combinational Logic, simulation and testing of logic circuits

Multiplexers, Decoders and Programmable Logic Devices: Multiplexers, three state buffers, decoders and encoders, Programmable Logic devices.

Textbook 1: Part B: Chapter 8, Chapter 9 (Sections 9.1 to 9.6)

Laboratory Component:

1. Given a 4-variable logic expression, simplify it using appropriate technique and realize the simplified logic expression using 8:1 multiplexer IC.
2. Design and implement code converter I) Binary to Gray (II) Gray to Binary Code

Teaching-Learning Process	1. Demonstration using simulator 2. Case study: Applications of Programmable Logic device 3. Chalk and Board for numerical

Module-4

Introduction to VHDL: VHDL description of combinational circuits, VHDL Models for multiplexers, VHDL Modules.

Latches and Flip-Flops: Set Reset Latch, Gated Latches, Edge-Triggered D Flip Flop 3,SR Flip Flop, J K Flip Flop, T Flip Flop.

Textbook 1: Part B: Chapter 10(Sections 10.1 to 10.3), Chapter 11 (Sections 11.1 to 11.7)

Laboratory Component:

1. Given a 4-variable logic expression, simplify it using appropriate technique and simulate the same in HDL simulator
2. Realize a J-K Master / Slave Flip-Flop using NAND gates and verify its truth table. And implement the same in HDL.

Teaching-Learning Process	1. Demonstration using simulator 2. Case study: Arithmetic and Logic unit in VHDL 3. Chalk and Board for numerical

Module-5

Registers and Counters: Registers and Register Transfers, Parallel Adder with accumulator, shift registers, design of Binary counters, counters for other sequences, counter design using SR and J K Flip Flops.

H. O. D.
Dept. Of Computer Science & Engineering
Alva's Institute of Engg. & Technology
Mijar, MOODBIDRI - 574 225