K.S. SCHOOL OF ENGINEERING AND MANAGEMENT, BENGALURU - 560109

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

CO-PO Mapping

Course: Data Structures and Applications				
Type: Integrated Professional Core Course			**Course Code: 21CS32**	
No of Hours				
Theory (Lecture Class)	Tutorials	Practical/Field Work/Allied Activities	Total/Week	Total hours of Pedagogy
4	0	3	7	40 T + 20 P

Marks			
CIE	SEE	Total	Credits
50	50	100	4

Aim/Objectives of the Course

1. To explain fundamentals of data structures and their applications essential for programming/problem solving.
2. To illustrate linear representation of data structures: Stack, Queues, Lists, Trees and Graphs.
3. To demonstrate sorting and searching algorithms.
4. To find suitable data structure during application development/Problem Solving.

Course Learning Outcomes

After completing the course, the students will be able to

CO1	**Apply** the basic data structures concepts such as arrays, structures, unions, pointers, strings and dynamic memory allocation function to solve simple problems.	**Applying (K3)**
CO2	**Make use of** stacks to evaluate mathematical expression and apply queues to solve problems.	**Applying (K3)**
CO3	**Utilize** linked list for implementation of lists, stacks, queues, polynomials and sparse matrix.	**Applying (K3)**
CO4	**Construct** various types of trees using linked list and array representation and apply tree traversal method for expression evaluation.	**Applying (K3)**
CO5	**Make use of** BFS, DFS, searching, sorting, hashing techniques appropriately.	**Applying (K3)**

Syllabus Content

Module 1: Introduction: Data Structures, Classifications (Primitive & Non Primitive), Data structure Operations, Review of Arrays, Structures, Self-Referential Structures, and Unions. Pointers and Dynamic Memory Allocation Functions. Representation of Linear Arrays in Memory, Dynamically allocated arrays. **Array Operations:** Traversing, inserting, deleting, searching, and sorting. Multidimensional Arrays, Polynomials and Sparse Matrices. **Strings:** Basic Terminology, Storing, Operations and Pattern Matching algorithms.	**CO1** 8 hrs PO1-1 PO2-3 PO3-3 PO4-3

Laboratory Experiments: 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) c. Display of Array Elements d. Exit. Support the program with functions for each of the above operations. **LO:** At the end of this session the student will be able to 1. Understand the basic data structures concepts. 2. Analyze the pattern matching problem and sparse matrix 3. Understand the string terminologies.	PO6-1 PO12 -1 PSO1-3 PSO2-1

Module 2: Stacks: Definition, Stack Operations, Array Representation of Stacks, Stacks using Dynamic Arrays, Stack Applications: Polish notation, Infix to postfix conversion, evaluation of postfix expression. **Recursion** - Factorial, GCD, Fibonacci Sequence, Tower of Hanoi, Ackerman's function. **Queues**: Definition, Array Representation, Queue Operations, Circular Queues, Circular queues using Dynamic arrays, dequeues, Priority Queues. **Laboratory Experiments:** 1. Design, Develop and Implement a menu driven Program in C for the following operations on STACK of Integers (Array Implementation of Stack with maximum size MAX) a. Push an Element on to Stack b. Pop an Element from Stack c. Demonstrate Overflow and Underflow situations on Stack d. Display the status of Stack e. Exit .Support the program with appropriate functions for each of the above operations 2. Design, Develop and Implement a Program in C for the following Stack Applications a. Evaluation of Suffix expression with single digit operands and operators: +, -, *, /, %, ^ b. Solving Tower of Hanoi problem with n disks. **LO:** At the end of this session the student will be able to 1. Analyze the stack operations. 2. Understand recursion concepts. 3. Define and solve the simple queue problems.	**CO2** 8 hrs. PO1-1 PO2-3 PO3-3 PO4-3 PO6-1 PO12-1 PSO1-3 PSO2-1
Module 3: Linked Lists: Definition, Representation of linked lists in Memory, Memory allocation; Garbage Collection. Linked list operations: Traversing, Searching, Insertion, and Deletion. Doubly Linked lists, Circular linked lists, and header linked lists. Linked Stacks and Queues. Applications of Linked lists – Polynomials, Sparse matrix representation. Programming Examples. **Laboratory Experiments:** 1. Singly Linked List (SLL) of Integer Data a. Create a SLL stack of N integer. b. Display of SLL c. Linear search. Create a SLL queue of N Students Data Concatenation of two SLL of integers. 2. Design, Develop and Implement a menu driven Program in C for the following operations on Doubly Linked List (DLL) of Professor Data with the fields: ID, Name, Branch, Area of specialization 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. **LO:** At the end of this session the student will be able to	**CO3** 8 hrs PO1-1 PO2-3 PO3-3 PO4-3 PO6-1 PO12-1 PSO1-3 PSO2-1

1. Understand the concepts of linked list. 2. Solve simple problems on linked list such as sparse matrix and polynomials.	
Module 4: Trees: Terminology, Binary Trees, Properties of Binary trees, Array and linked Representation of Binary Trees, Binary Tree Traversals - Inorder, postorder, preorder; Additional Binary tree operations. Threaded binary trees, Binary Search Trees – Definition, Insertion, Deletion, Traversal, Searching, Application of Trees-Evaluation of Expression. **Laboratory Experiments:** 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}. 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 **LO:** At the end of this session the student will be able to 1. Understand the tree terminologies. 2. Solve binary tree traversals. 3. Evaluate the expression of the given tree. 4. Determine the various operations on trees like insertion, deletion.	**CO4** 8 hrs PO1-1 PO2-3 PO3-3 PO4-3 PO6-1 PO12-1 PSO1-3 PSO2-1
Module 5: Trees 2: AVL tree, Red-black tree, Splay tree, B-tree. **Graphs**: Definitions, Terminologies, Matrix and Adjacency List Representation Of Graphs, Elementary Graph operations, Traversal methods: Breadth First Search and Depth First Search. **Hashing**: Hash Table organizations, Hashing Functions, Static and Dynamic Hashing. **Laboratory Experiments:** 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. **LO:** At the end of this session the student will be able to 1. Understand the graph terminologies. 2. Solve tree traversals using BFS & DFS methods. 3. Understand hashing technique. 4. Define the basics of file and their organization.	**CO5** 8hrs PO1-1 PO2-3 PO3-3 PO4-3 PO6-1 PO12-1 PSO1-3 PSO2-1
Text Books 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 (specify minimum two foreign authors text books)

1. Gilberg & 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.

Useful Websites

- https://nptel.ac.in/courses/106102064/
- https://www.youtube.com/watch?v=Db9ZYbJONHc
- https://www.youtube.com/watch?v=DFpWCl_49i0
- https://www.youtube.com/watch?v=3hyxc4juJRg

Useful Journals

- IEEE TECHNOLOGY NAVIGATOR
- Journal of informatics and data mining
- Journal of computer and system sciences-Elsevier

Teaching and Learning Methods

1. Lecture class: 40 hrs
2. Tutorial classes: 15 hrs
3. Practical classes: 20hrs

Assessment

Type of test/examination: Written examination

Continuous Internal Evaluation(CIE) : 1) Three Tests each of 20 marks (duration 01 hour)

2) Two assignments each of 10 Marks

3) Practical Sessions for 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

Total CIE: 50 Marks

Semester End Exam (SEE) : 100 marks (students have to answer all main questions) which will be reduced to 50 Marks.

Test duration: 1 hr

Examination duration: 3 hrs

CO to PO Mapping

PO1: Science and engineering Knowledge **PO2:** Problem Analysis **PO3:** Design & Development **PO4:**Investigations of Complex Problems **PO5:** Modern Tool Usage **PO6:** Engineer & Society	**PO7:**Environment and Society **PO8:**Ethics **PO9:**Individual & Team Work **PO10:** Communication **PO11:**Project Mngmt & Finance **PO12:**Life long Learning

PSO1: An ability to design and develop Artificial Intelligence technology into innovative products for solving real world problems.

PSO2: An ability to design and develop Data Science methods for analyzing massive datasets to extract insights by applying AI as a tool.

CO	PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO 1	PSO 2
18CS 32	K-level														
CO1	K3	1	3	3	3	-	1	-	-	-	-	-	1	3	1
CO2	K3	1	3	3	3	-	1	-	-	-	-	-	1	3	1
CO3	K3	1	3	3	3	-	1	-	-	-	-	-	1	3	1
CO4	K3	1	3	3	3	-	1	-	-	-	-	-	1	3	1
CO5	K3	1	3	3	3	-	1	-	-	-	-	-	1	2	1

Course In charge HOD-AI & DS IQAC Coordinator Principal