DAYANANDA SAGAR COLLEGE OF ENGINEERING

An Autonomous Institute Affiliated to VTU, Belagavi Approved by AICTE; ISO 9001:2015
Certified Accredited by National Assessment Accreditation Council (NAAC) with 'A' grade
Shavige Malleshwara Hills, Kumaraswamy Layout, Bengaluru-560078

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

COURSE HANDOUT

Academic Year	**2022-2023**	**L: T: P: S:**	**3:0:1:0**
Course Title	**Data Structures & its applications**	**Course code**	**22CS33**
Credit Units	**04**	**Course Type**	**Integrated**
Course Authors	**Prof. A M Prasad & Prof. Aruna S**	**Date Approved**	**25/09/2023**
Exam Hours	**03**	**CIE Marks**	**50**
Total Hours:	**52 [T:Theory 40 + L:Laboratory 12]**	**SEE Marks**	**50**

Course objectives:

1. Enlighten on fundamentals of data structures, their applications in real life.
2. Apply data structures such as stack, queue and linked list for the given problem.
3. Assess appropriate data structure during program development / problem solving.

Course Outcomes [CO]: At the end of the course, student will be able to:

CO1	Understand and explore the concepts of data structures to solve problems.
CO2	Analyze time and space efficiency of algorithms
CO3	Apply the concept of linear and non-linear data structures to various applications.
CO4	Analyze the usage of precise data structure for a given application.
CO5	Design and implement the operations on various data structures.
CO6	Use Data Structures in high level programming language like C/C++.

Unit	**Contents**	**Hours**	**COs**
Unit – 1	**Data Structures:** Classifications Primitive and Non Primitive, Data structure Operations, Review of Arrays, Structures and Unions, Self-Referential Structure. Pointers and Dynamic Memory Allocation Functions. Dynamically allocated arrays. Algorithm specification, performance analysis & measurements. **Stack:** Definition, Stack Operations, Array Representation of Stacks, Stacks using Dynamic Arrays. Stack Applications: Infix to postfix conversion, Evaluation of postfix expression, Recursion - Factorial, GCD, Fibonacci Sequence and Tower of Hanoi.	08	**CO1 & CO2**
	Hands-on: Stack applications: conversion of mathematical expressions, Recursion: Tower of Hanoi. **Case study on stack: Function execution, Nested functions**	02	
Unit - 2	**Queues:** Definition, Array Representation, Queue Operations, Circular Queues, Circular queues using Dynamic arrays, Dequeues, Priority Queues. Multiple Stacks and Queues. **Linked Lists:** Definition, Representation of linked lists in Memory, Memory allocation, Singly Linked list operations: Traversing, Searching, Insertion, and Deletion.	08	**CO1 & CO2**
	Hands-on: Queue Applications: Printer spooling, Linked list Applications: Polynomial Addition	02	

	Case Study on queue: Josephus problem, CPU scheduling.		
Unit – 3	**Advanced Linked Lists:** Doubly Linked lists, Circular linked lists, and header linked lists. Linked Stacks and Queues. Applications of Linked lists – Polynomials, Sparse matrix representation. **Trees:** Terminology, Binary Trees, Properties of Binary trees, Array and linked Representation of Binary Trees, Binary Tree Traversals - Inorder, postorder, preorder.	08	**CO2 & CO3**
	Hands-on: Doubly linked list Applications: Creation of Data structures, Binary tree operations: Conversion of expressions. **Case Study on linked list : Text Editor, Assembler - Creation of a Symbol Table.**	03	
Unit – 4	**Binary Search Trees** – Definition, Insertion, Deletion, Traversal, Searching, Application of Trees-Evaluation of Expression. B-Trees: Definition of B-trees Basic operations. **Efficient binary search trees:** AVL Trees, Red black Tree, Optimal Binary Search Tree	08	**CO1 & CO3**
	Hands-on: BST operations: Searching and Sorting, AVL tree: Efficient search and Sort. **Case Study on B trees: Indexing in databases.**	03	
Unit - 5	**Graph:** Elementary Graph operations, Representation of computer network topology, Traversal methods: Breadth First Search and Depth First Search. **Hashing:** Table Representation: hash functions, collision resolution separate chaining, open addressing-linear probing, quadratic probing, double hashing.	08	**CO1 & CO4**
	Hands-on: Application of BFS and DFS: Connectivity of graph, finding path in a computer network. **Case Study on hashing: URL Decoding, password search, Spanning Trees.**	02	

Text Books:

Sl.#	Author & Edition	Publisher
1.	Horowitz, Sahni, Anderson-Freed: Fundamentals of Data Structures in C, 2nd Edition	Universities Press, 2007.
2.	Data Structures using C and C++, Yedidyah Langsam Moshe J. Augenstein and Aaron M. Tenenbaum, 2nd Edition	PHI/Pearson, 2009

Reference Books:

Sl.#	Author & Edition	Publisher
1.	Richard F. Gilberg and Behrouz A. Forouzan: Data Structures A Pseudocode Approach with C	Cengage Learning, 2005.
2.	A.M Padma Reddy," Approach of Data Structures" , 5th Edition	Person Publication, 5th Edition, 2015
3.	Data Structures and Algorithms in Java By Robert Lafore ·	2017

E-Learning:

1.	A course on Data Structures and Algorithms (Harvard CS 124) offered by Prof. Madhu Sudan at Harvard - https://docs.google.com/spreadsheets/d/1pLWwZaHUWXFAUjivWCi6go96nd3-nAFzc4XvZKNI2co/edit
2.	A course on Data Structures on Coursera - https://www.coursera.org/learn/data-structures?specialization=data-structures-algorithms

Self-Study Component:

Reference Books: Introduction to Algorithms, Charles E. Leiserson Ronald L. Rivest Clifford Stein, 4th edition, MIT Press.

Unit	Prerequisite [Studied in syllabus]	Topic (Self-Study)	Reference link
Unit – 1	Stacks	Amortized Analysis	https://youtu.be/d605guaOH3A?si=TT0UmNUZffldORNV
Unit – 2	Queues	Heap Sorting	https://builtin.com/data-science/heap-sort
Unit – 3	Linked-list	Linked-list representation of disjoint sets	https://youtu.be/a59_5VrSFWM
Unit – 4	B-Trees, Red-Black Trees	Fibonacci Heaps An interval tree	https://youtu.be/6JxvKfSV9Ns?si=R0aS3pmfySbmpMSj
Unit - 5	Hashing	Cryptographic hashing	https://youtu.be/gTfNtop9vzM?si=sdHol5JXERbantWu

Assignment Scheme [Certification Unit-wise]

#	Topics	Infosys springboard Link	Marks	Total
a.	Data Structures Stack and Queue	https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_01331151626894540811 9_shared/overview	02	10
b.	Data Structures Linked Lists	https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_0133115155445514 24117_shared/overview	02	
c.	Introduction to Binary Trees & Sorting	https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_01350157816505139210584/overview	02	
d.	Red-Black Trees	https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_01330153829574246426450_shared/overview	02	
e.	Introduction to Graphs & Hashed Data Structures	https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_01350157311076761610332/overview	02	

Alternative Assignment Tool [Quiz, online mode]

Unit	Number of Multiple choice questions	Marks	Total
Unit – 1	06	06	Scale down to 10
Unit – 2	06	06	10
Unit – 3	06	06	
Unit – 4	06	06	
Unit - 5	06	06	

Continuous Internal Assessment

CIA#	Marks	Portions

CIA 1	30	Unit – 1 , Unit – 2
CIA 2	30	Unit – 3, Unit – 4
CIA 3	30	Unit – 5

Integrated Laboratory Pre-Requisites:

1.	Certification: Pointers In C Programming
2.	Course Time: 01 hr 45 mins, Advanced level. Student can earn a certificate on completing the content https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_01384203240484864010470_shared/overview

Integrated Laboratory Objectives and Outcomes:

Objectives:

1. Introduce the concept of data structures through ADT including List, Stack, Queues
2. To design and implement various data structure algorithms.
3. To introduce various techniques for representation of the data in the real world
4. To develop application using data structure algorithms

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

CO1	Select appropriate data structures to be used in the programming solution for a given problem.
CO2	Implement operations like searching, insertion, and deletion, traversing mechanism etc. on various data structures.
CO3	Implement Linear and Non-Linear data structures efficiently.
CO4	Implement various operations on Linked Lists
CO5	Design and Implement applications of Non-linear data structure.
CO6	Implement graph traversal algorithms

The student should be able to:

1	Analyze and Compare various linear and non-linear data structures
2	Code, debug and demonstrate the working nature of different types of data structures and their applications
3	Implement, analyze and evaluate the searching and sorting algorithms
4	Choose the appropriate data structure for solving real world problems

Laboratory exercises

Q.no	Problem Statement
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 how Stack can be used to check Palindrome d. Demonstrate Overflow and Underflow situations on Stack e. Display the status of Stack f. Exit. Support the program with appropriate functions for each of the above operations
2	Design, Develop and Implement a Program in C for converting an Infix Expression to Postfix Expression. Program should support for both parenthesized and free parenthesized expressions with the operators: +, -, *, /, % (Remainder), ^ (Power) and alphanumeric operands.
3	Design, Develop and Implement a Program in C for the following Stack Applications. Evaluation of Suffix expression with single digit operands and operators: +, -, *, /, %, ^
4.	Design, Develop and Implement a menu driven Program in C for the following operations on Circular QUEUE of Characters (Array Implementation of Queue with maximum size MAX)

	a. Insert an Element on to Circular QUEUE b. Delete an Element from Circular QUEUE c. Demonstrate Overflow and Underflow situations on Circular QUEUE d. Display the status of Circular QUEUE e. Exit Support the program with appropriate functions for each of the above operations
5.	Design, Develop and Implement a menu driven Program in C for the following operations on Singly Linked List of Student Data with the fields: USN, Name, Programme, Sem, PhNo a. Create a SLL of N Students Data by using front insertion. b. Display the status of SLL and count the number of nodes in it c. Perform Insertion/Deletion at End of SLL d. Perform Insertion/Deletion at Front of SLL e. Exit
6.	Design, Develop and Implement a menu driven Program in C for the following operations on Doubly Linked List (DLL) of Employee Data with the fields: SSN, Name, Dept, Designation, Sal, PhNo a. Create a DLL of N Employees Data by using end insertion. b. Display the status of DLL and count the number of nodes in it c. Perform Insertion and Deletion at End of DLL d. Perform Insertion and Deletion at Front of DLL e. Demonstrate how this DLL can be used as Double Ended Queue. f. Exit
7.	Design, Develop and Implement Program in C for the following operations on Singly Circular Linked List (SCLL) with header nodes a. Represent and Evaluate a Polynomial $P(x,y,z) = 6x^2y^2z - 4yz^5+3x^3yz+2xy^5z- 2xyz^3$ b. Find the sum of two polynomials POLY1(x,y,z) and POLY2(x,y,z) and store the result in POLYSUM(x,y,z) Support the program with appropriate functions for each of the above operations
8.	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 c. Search the BST for a given element (KEY) and report the appropriate message d. Exit
9.	Design, Develop and Implement a Program in C for the operations on Graph(G) of Cities a. Create a Graph of N cities using Adjacency Matrix. b. Print all nodes reachable from a given source node in a graph using DFS/BFS method
10.	Given a File of N employee records with a set K of Keys (4-digit) which uniquely determine the records in file F. Assume that file F is maintained in memory by a Hash Table (HT) of m memory locations with L as the set of memory addresses (2-digit) of locations in HT. Let the keys in K and addresses in L are Integers. Design and develop a Program in C that uses Hash function H: K→L as H (K) =K mod m, and implement hashing technique to map a given key K to the address space L. Resolve the collision if any using linear probing.

Evaluation Scheme [Integrated]

Theory				
Sl.no	**Particulars**	**Max. Marks**	**Scale down to 10**	**Max. Marks**
a.	Continuous Internal Assessment -1	30	Average of 03 Assessments	10
b.	Continuous Internal Assessment -2	30		
c.	Continuous Internal Assessment -3	30		
d.	Assignment [Problem Based]	10	10	10
e.	Alternative Assessment	30	10	10
			Total	**30**
			Minimum Requirement	**12**

2. Laboratory				
Sl.no	**Particulars**	**Max. Marks**	Scale down to 20	**Max. Marks**
a.	Continuous Internal Evaluation	30		**20**
b.	Laboratory Internal Examination	50		

Minimum requirement	**08**

NOTE: The student is not allowed take examination, if fails to score minimum requirement marks either in theory or laboratory