GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name:Engineering

Level: Diploma

Branch: Computer Engineering / Computer Science & Engineering

Subject Code : DI03000021

Subject Name : Data Structures

W.E. F. Academic Year:	2024-25
Semester:	3rd
Category of the Course:	PCC

Prerequisite:	Basic knowledge of Programming language and basic mathematics
Rationale:	The Data Structures curriculum is designed to build problem-solving and algorithmic thinking by teaching fundamental data structures (arrays, linked lists, stacks, queues, trees, graphs) and their real-world applications. It emphasizes efficiency analysis, hands-on implementation using programming language such as C/C++/Python programming and industry relevance through coding projects and competitive programming. The structured approach ensures progressive learning, while self-learning resources (MOOCs, coding platforms) enhance student engagement. This course prepares students for software development, system design, and technical interviews, making them academically and professionally ready.

Course Outcome:

After Completion of the Course, Student will able to:

No	Course Outcomes	RBT Level
01	Interpret the concepts of Data Structure, Algorithm and Array.	Understand
02	Implement various algorithms of stack and queue data structures.	Apply
03	Apply basic operations on the linked list data structure.	Apply
04	Illustrate the use of Tree, Binary Search Tree and Graph.	Apply
05	Apply different sorting and searching algorithms on data sets.	Apply

**Revised Bloom's Taxonomy (RBT)*

GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name:Engineering

Level: Diploma

Branch: Computer Engineering / Computer Science & Engineering

Subject Code : DI03000021

Subject Name : Data Structures

Teaching and Examination Scheme:

Teaching Scheme (in Hours)			Total Credits L+T+ (PR/2)	Assessment Pattern and Marks				Total Marks
				Theory		Tutorial / Practical		
L	T	PR	C	ESE (E)	PA(M)	PA(I)	ESE (V)	
3	0	2	4	70	30	20	30	150

Course Content:

Unit No.	Content	No. of Hours	% of Weightage
1.	**Basic Concepts of Data Structures** Data, Data Structure, Types of data structures, Primitive and Non-Primitive, Linear and Non-Linear data structures, Introduction to Algorithms, Characteristics of an Algorithm, Time Complexity, Space Complexity, Asymptotic Notations: Big Oh Notation, Big Omega Notation, Big Theta Notation. **Array:** Array Representation, Row Major Arrays**,** Column Major Arrays**,** Array operations: Insert, Update, Delete, Traverse, Search	**06**	**14**
2.	**Stack and Queues** **Stack:** Array representation of Stack, PUSH - POP Operations on Stack, Implementation of Stack, Applications of Stack, Infix, Prefix and Postfix Forms of Expressions with example, Recursive Functions - Factorial, Greatest Common Divisor, Fibonacci series **Queue:** Array representation of Queue, Insert and Delete Operations on Queue, Implementation of Queue, Limitation of Simple Queue, Concepts of Circular Queue, Applications of queue, Differentiate simple queue and circular queue	**08**	**18**

3.	**Linked List** Pointer, Structure, Structure using pointer, Dynamic Memory Allocation **Linked List:** Linked List representation, Types of Linked List- Singly linked list, Circular linked list, Doubly linked list, Applications of linked list **Basic operations on Singly linked list**: Insertion of a new node in the beginning and end of the list, Insertion of a new node before/after a given node, Insertion of a new node in sorted linked list, Deleting the first and last node from a linked list, searching a node in Linked List, Count the number of nodes in linked list	**11**	**24**
4.	**Trees and Graph** **Tree**: Fundamental concepts - General Tree, Forest, Root node, Parent node, Sibling, Leaf node, Degree, In-degree, Out-degree, Sub tree, Level, Path, Depth, Height, Binary tree, Complete Binary tree, Strict Binary tree, Conversion of General Tree to Binary Tree, Tree Traversal: Inorder, Preorder, Postorder **Binary Search Tree (BST):** Insertion, Deletion and Searching of a node in BST. **Graph**: Fundamental concepts - Vertex, Edge, Path, Cycle, Types of Graphs: Directed Graph and Undirected Graph, Connected Graph, Complete Graph, Weighted Graph	**09**	**20**
5.	**Sorting and Searching** **Sorting:** Bubble Sort, Selection Sort, Insertion Sort, Quick Sort, Merge Sort, Radix Sort algorithms with example. **Searching:** Linear Search and Binary Search algorithms with example. **Hashing:** Key, Hash table, Hash functions: Division Method, Mid Square Method, Folding Method.	**11**	**24**
	Total	**45**	**100**

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks (in %)					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
20	**38**	**42**	**0**	**0**	**0**

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

References/Suggested Learning Resources:

(a) Books:

1. Jean-Paul Tremblay & Paul G. Sorenson "An Introduction to Data Structures with Applications", Tata McGraw Hill
2. Tenen Baum "Data Structures using C & C++", Prenctice-Hall International
3. Thareja, Reema "Data and File Structures using C", Oxford University Press New Delhi 2011
4. G. S. Baluja "Data Structures Through C (A Practical Approach)", Dhanpat Rai & Co. Ltd, 2015
5. Samanta, D. "Classic Data Structures", PHI Learning, New Delhi
6. Chitra, A Rajan, P T "Data Structures", Tata McGraw Hill, New delhi, 2009

(b) Open source software and website:

1. https://www.programiz.com/dsa
2. https://nptel.ac.in/courses/106102064
3. https://nptel.ac.in/courses/106106133
4. https://www.codecademy.com/learn/linear-data-structures
5. https://www.tutorialspoint.com/data_structures_algorithms/data_structures_basics.htm
6. https://www.udacity.com/course/data-structures-and-algorithms-in-python--ud513
7. https://www.edx.org/learn/data-structures
8. https://www.geeksforgeeks.org/dsa-tutorial-learn-data-structures-and-algorithms/
9. https://www.w3schools.com/dsa/dsa_intro.php

Suggested Course Practical List:

1. Create a program to find maximum out of 10 numbers by using array and find time complexity of that program.
2. Implement push and pop algorithms of stack.
3. Implement recursive functions (Factorial).
4. Implement insert and delete algorithms of queue.
5. Implement simple structure programs using pointers.
6. Implement insertion a node algorithm in singly linked list.
7. Implement delete a node algorithm in singly linked list.

8. Implement counting no of nodes algorithm in singly linked list.
9. Implement searching of a node algorithm in singly linked list.
10. Implement construction of binary search tree.
11. Implement Bubble sort algorithm.
12. Implement Selection sort algorithm.
13. Implement Insertion sort algorithm.
14. Implement Binary search algorithm.

List of Laboratory/Learning Resources Required:

1. Computer with basic configuration with windows or unix os
2. C/Python Compiler

Suggested Project List:

1. Develop a C/Python Program that evaluate the given arithmetic expression using stack.
2. Develop a C/Python Program that maintain a queue of persons. In this queue user can add a person, delete a person and search a person.
3. Develop a C/Python Program that perform banking operations like withdraw cash, deposit cash and mini statement using appropriate data structure.
4. Develop a C/Python Program for process management algorithm by using appropriate data structure.
5. Develop a C/Python Program for print spooler using appropriate Data structure.
6. Develop a C/Python Program for Telephone Directory system. In this user can adding, searching, modifying, listing, and deleting records through the use of appropriate data structure.

Suggested Activities for Students:

a) Encourage students to participate in different coding competitions like Hackathon, online competitions on code chef etc.
b) Students are encouraged to register themselves in various MOOCs such as: Swayam, edx, Coursera, Udemy etc to further enhance their learning.
c) Prepare charts to explain use/process of the identified topic.
d) https://www.codechef.com/, in this website very elementary programs are available, students are expected to solve those programs
e) Encourage students to form a coding club at institute level.
f) https://code.org/, an hour of coding event may be organized and students are encouraged to participate.