GUJARAT TECHNOLOGICAL UNIVERSITY

INSTRUMENTATION AND CONTROL (APPLIED INSTRUMENTATION) (03)

DATA STRUCTURE WITH OBJECT ORIENTED PROGRAMMING

SUBJECT CODE: 2710314

SEMESTER: I

Type of course: Major Elective: 1

Prerequisite: C/C++ programming

Rationale: This course covers good principles of algorithm design and fundamental data structures. The emphasis is on choosing appropriate data structures to represent real world problems and designing correct and efficient algorithms to operate on these data structures and solve the problems.

Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks						Total Marks
L	T	P	C	Theory Marks		Practical Marks				
				ESE (E)	PA (M)	PA (V)		PA (I)		
						ESE	OEP	PA	RP	
3	2	2	5	70	30	20	10	20	0	150

Content:

Sr. No.	Topics	Teaching Hrs.	Module Weightage
1	Introduction to Data Structures: Data types – primitive and non-primitive, Types of Data Structures- Linear & Non Linear Data Structures.	01	2%
2	**Linked Lists:** Singly Linked Lists, Doubly Linked Lists, Circular lists, Skip Lists, Self-Organizing Lists, Sparse Tables, Polynomials-Representation, adding polynomials, Erasing Polynomials, Circular List Representation of polynomials	05	13%
3	**Stacks and Queues:** Stacks, Queues, Priority Queues, Evaluation of Expressions-Expressions, Postfix Notation, Infix to Postfix	04	10%
4	**Recursion:** Function Calls and Recursive Implementation, Anatomy of a Recursive Call, Tail Recursion, Nontail Recursion, Indirect Recursion, Nested Recursion, Excessive Recursion	04	10%
5	**Binary Trees:** Trees, Binary Trees, and Binary Search Trees, Implementing Binary Trees, Searching a Binary Search Tree, Tree Traversal- Breadth First Traversal, Depth-First Traversal, Stackless Depth-First Traversal, Insertion, Deletion, Balancing a Tree- DSW Algorithm, AVL Trees, Self-Adjusting Trees- Self-Reconstructing Trees, Splaying, Heaps, Polish Notation and Expression Trees, Application: Huffman Algorithm	15	30%
6	**Graphs:** Graph Representation, Graph Traversals, Shortest Paths, Cycle Detection, Spanning Trees, Connectivity in Undirected Graphs, Connectivity in Directed Graphs, Topological Sort, Networks	06	10%
7	**Sorting:** Insertion Sort, Selection Sort, Bubble Sort, Comb Sort, Decision Trees, Shell Sort, Heap Sort, Quicksort, Mergesort, Radix Sort, Counting Sort	06	15%
8	**Hashing:** Hash Functions-Division, Folding, Mid-Square Function,	04	10%

	Extraction, Radix Transformation, universal Hash Functions, Collision Resolution- Open Addressing, chaining, Bucket Addressing, Deletion		

Reference Books:

1. Adam Drozdek, “ Data Structures and Algorithms in C++”, Cengage Learning
2. Ellis Horowitz, Sartaj Sahni and Dinesh Mehta, “Fundamentals of Data Structures in C++”, Galgotia Publication
3. Langsam, Auenstein and Tenenbaum, “ Data Structures Using C and C++”, PHI Publication
4. Sartaj Sahni, “Data Structures, Algorithms and Applications in C++”, Universities Press
5. Balagurusamy,” Object oriented programming with C++”, Tata McGraw-Hill
6. Robert Lafore, “Object Oriented Programming in C++”, Pearson Education

Course Outcome:

After learning the course the students should be able to

1. Design an appropriate data structure for modeling given computing problems.
2. Apply appropriate for solving given computing problems.
3. Understand and implement the both array based and linked-list based data structures, including singly, doubly, and circular linked-lists.
4. Understand and implement the Stack data structure and stack operations.
5. Understand and implement the both array based circular queue and linked-list based queue implementations.
6. Understand and implement general tree data structures, including binary tree, both array based and reference based implementations.
7. Understand and implement binary search trees.
8. Understand and implement different sorting algorithms
9. Understand and implement graph data structures.

Have practical knowledge on the application of data structures

List of Experiments:

Student has to prepare programs for various data structures covered in this course with any soft computing tools (C, C++).

Major Equipments:

Computer Laboratory

Open Ended Problem:

1. Simulate a simple dictionary. Assume each character contains at least 10 vocabularies. Create an index page for all characters. Retrieve the word using index value. Assume that the index characters from a to z.
2. Design a simple search engine to display the possible websites upon entering a search query. Use suitable data structure for storage and retrieval.
3. Design and Develop the index for a text book of at least 100 pages using alphabets.
4. Design a Student Prerequisite Subjects Management System requires the use of linked list or tree to store different courses and their prerequisites and based on this list it will allow any student to take any course or not.
5. Write a program that uses the radix sort to sort 1000 random digits. Print the data before and after the sort. Each sort bucket should be a linked list. At the end of the sort, the data should be in the original array.

List of Open Source Software/learning website:

- C, C++
- NTPEL