

GUJARAT TECHNOLOGICAL UNIVERSITY, AHMEDABAD, GUJARAT

Course Curriculum

DATA STRUCTURES (Code: 3330704)

Diploma Programme in which this course is offered	Semester in which offered
Computer Engineering, Information Technology	3 rd Semester

1. RATIONALE

Development of application systems and software that use underlying architecture of machines efficiently and effectively requires the ability to use and manipulate various types of Data Structures and other constructs. This being a fundamental ability which is language neutral, yet requires use of a language for its implementation. This is a basic course which goes along with other programming courses to develop an integrated ability to efficient software development, hence this course is very important for computer and IT diploma engineers.

2. COMPETENCY (Programme Outcome according to NBA Terminology):

The course content should be taught and implemented with the aim to develop various types of skills so that students are able to acquire following competency:

- **Implement various types of algorithms using Data Structures.**

3. TEACHING AND EXAMINATION SCHEME

Teaching Scheme (In Hours)			Total Credits (L+T+P)	Examination Scheme				Total Marks
				Theory Marks		Practical Marks		
L	T	P	C	ESE	PA	ESE	PA	150
3	2	2	7	70	30	20	30	

Legends: L -Lecture; T -Tutorial/Teacher Guided Student Activity; P -Practical; C - Credit;ESE-End Semester Examination; PA -Progressive Assessment

4. COURSE DETAILS

Unit	Major Learning Outcomes (Course Outcomes in Cognitive Domain according to NBA terminology)	Topics and Sub-topics
Unit – I Basic Concepts of Data Structures	1a. Represent the data in relevant memory	1.1 Data Structure Basic Concepts 1.2 Types of data structures
	1b. Differentiate primitive and non-primitive data structures	1.3 Primitive and non-primitive data structures
	1c. List key features of an algorithm	1.4 Introduction to Algorithms 1.5 Key features of an algorithm
	1d. Define time complexity and space complexity	1.6 Analysis Terms (for the definitions purpose only) : a. Time Complexity b. Space Complexity c. Asymptotic Notations ,Big ‘O’, Notation , Best case Time Complexity, Average case Time Complexity, Worst case Time Complexity
	1e. Design and Implement programs to represent array in row major and column major order	1.7 Array : i. Row Major Arrays ii. Column Major Arrays 1.8 Overview of various array operations.
	1f. Design and Implement search algorithms	1.9 Searching an element into an array – i. Linear Search ii. Binary Search
Unit– II Strings	2a. Create strings	2.1 String representation : Reading and Writing Strings
	2b. Develop algorithms to implement various operations on string	2.2 String operations : Finding length of a string, Converting Characters of a string into upper case and lower case, Concatenation of two strings to form a new string, Appending, Reversing a string, Copying a string, Comparing strings, Insertion, Substring, Deletion
Unit– III Stack and Queues	3a. Define linear and non-linear data structures and develop algorithms to push an element into stack, pop an element from the stack.	3.1 Linear and Non-Linear Data Structures 3.2 Stack : Array representation of Stack, PUSH POP Operations on Stack, Implementation of Stack, Application of Stack, Infix, Prefix and Postfix Forms of Expressions, Recursive Functions (factorial, greatest common divisor, Fibonacci series)

Unit	Major Learning Outcomes (Course Outcomes in Cognitive Domain according to NBA terminology)	Topics and Sub-topics
	3b. Implement Queue with various operations like insert delete	3.3 Queue: Array representation of Queue, Operations on Queue, Implementation of Queue, Limitation of Single Queue
	3c. Differentiate circular and simple queue	3.4 Concepts of Circular Queue 3.5 Application of queue 3.6 Difference circular queue and simple queue
Unit– IV Linked List	4a. Define linked list	4.1 Pointers Revision 4.2 Revision of Structure 4.3 Revision of structure using pointers 4.4 Dynamic Memory Allocation 4.5 Linked list Presentation 4.6 Types of Linked List
	4b. Develop algorithms to insert node from front, to the end, at any position, delete element, insert into sorted list, delete node from singly linked list	4.7 Basic operations on singly linked list : Insertion of a new node in the beginning of the list, at the end of the list, after a given node, before a given 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
	4c. Distinguish circular linked list and singly linked list	4.8 Concepts of circular linked list 4.9 Difference between circular linked list and singly linked list
	4d. Develop algorithms to insert node from front, to the end, at any position, delete node from doubly linked list	4.10 Basic operations on Doubly linked list : Insertion of a new node in the beginning of the list, at the end of the list, after a given node, before a given node. Deleting the first and last node from a linked list, Searching a node in Linked List, Count the number of nodes in linked list
	4e. List the applications of linked list	4.11 Applications of linked list
Unit– V Sorting and Hashing	5a. Arrange data in ascending and descending orders using appropriate sorting algorithm	5.1.Sorting Methods : a. Bubble Sort, b. Selection Sort, c. Quick Sort, d. Insertion Sort, e. Merge Sort, f. Radix Sort
	5b. Apply various hashing techniques	5.2.Hashing Concepts 5.3.Hash functions : Division Method, Middle Square Method, Folding Method,

Unit	Major Learning Outcomes (Course Outcomes in Cognitive Domain according to NBA terminology)	Topics and Sub-topics
	5c. Apply collision resolution techniques	5.4.Collision in Hashing 5.5.Collision resolution techniques: Linear Probing
Unit– VI Trees	6a Define non-linear data structure	6.1 Non-linear data structures
	6b Develop algorithms to manipulate tree	6.2 Binary trees : Complete Binary Tree, Basic Terms: level number, degree, in-degree and out-degree, leaf node, similar binary trees, copies of binary trees, directed edge, path, depth, General Tree, Conversion of General Tree to Binary Tree, Forest.
	6c Implement various tree manipulation algorithms	6.3 Binary Search Tree : Insertion of a node in binary tree, Deletion of a node in binary tree, Searching a node in binary tree 6.4 Tree Traversal : Inorder, Preorder, Postorder
	6d List applications of tree	6.5 Applications of binary tree

5. SUGGESTED SPECIFICATION TABLE WITH HOURS & MARKS (THEORY)

Unit	Unit Title	Teaching Hours	Distribution of Theory Marks			
			R Level	U Level	A Level	Total Marks
I	Basic Concepts of Data Structures	4	4	3	0	7
II	Strings	4	2	2	3	7
III	Stack and Queues	8	2	6	6	14
IV	Linked List	10	4	8	2	14
V	Sorting and Hashing	8	2	6	6	14
VI	Trees	8	4	4	6	14
Total		42	18	29	23	70

Legends: R = Remember; U= Understand; A= Apply and above levels (Bloom's revised taxonomy)

Note: This specification table shall be treated as a general guideline for students and teachers. The actual distribution of marks in the question paper may slightly vary from above table.

6. SUGGESTED LIST OF EXERCISES/PRACTICAL

The practical/exercises should be properly designed and implemented with an attempt to develop different types of practical skills (**Course Outcomes in psychomotor and affective domain**) so that students are able to acquire the competencies (Programme Outcomes). Following is the list of practical exercises for guidance.

Note: Here only Course Outcomes in psychomotor domain are listed as practical/exercises. However, if these practical/exercises are completed appropriately, they would also lead to development of **Programme Outcomes/Course Outcomes in affective domain** as given in a common list at the beginning of curriculum document for this programme. Faculty should refer to that common list and should ensure that students also acquire those Programme Outcomes/Course Outcomes related to affective domain.

Sr. No.	Unit No.	Practical Exercises (Course Outcomes in Psychomotor Domain according to NBA Terminology)	Approx. Hrs. Required
1	I	Define various terms such as algorithm, various approaches to design an algorithm, time complexity, space complexity, big 'o' notation, best case, average case and worst case time complexity etc. Develop simple program using pointer to a structure	2
2	I	Implement array using row major order and column major order	2
3	I	Implement Sequential search algorithms	2
4	I	Implement Binary search algorithms	2
5	II	Implement various string algorithms	2
6	III	Implement push and pop algorithms of stack using array	2
7	III	Implement recursive functions	2
8	III	Implement insert, delete algorithms of queue using array	2
9	III	Implement insert, delete algorithms of circular queue	2
10	IV	Implement simple structure programs using pointers	2
11	IV	Implement insertion of node in the beginning of the list and at the end of list in singly linked list	4
12	IV	Implement insertion of node in sorted linked list	2
13	IV	Implement insertion of node at any position in linked list	2
14	IV	Implement searching of a node, delete a node and counting no of node algorithms in singly linked list	4
15	IV	Implement insertion of node in the beginning and at the end of doubly linked list	2
16	IV	Implement insertion of node at any position in doubly linked list	2
17	IV	Implement searching of a node, delete a node and counting no of node algorithms in doubly linked list	4
18	V	Implement Bubble sort, Selection sort algorithms	2
19	V	Implement Quick Sort	2
20	V	Implement Insertion sort, Shell sort algorithm	2

21	V	Implement Merge Sort algorithm	2
22	V	Solve hash table example using division method, method square method, folding method (paper work only)	2
23	VI	Implement construction of binary search tree	2
24	VI	Implement inorder, preorder and postorder traversal methods in binary search tree	2
25	VI	Implement searching algorithm in binary search tree	2
Total			56

Note: In tutorials - Students will write programs and in practical session -execute program

7. SUGGESTED LIST OF STUDENT ACTIVITIES

Following is the list of proposed student activities like:

Prepare seminars on various topics like stack, queue, singly linked list algorithms, doubly linked list, sorting algorithms, tree algorithms etc.

Prepare diagrams/chart to display various sorting methods, string representation; data representations etc.

Develop a program with real life application example of particular topic.

Compare various sorting and searching methods. Prepare hash tables using given methods.

8. SPECIAL INSTRUCTIONAL STRATEGIES (If Any)

The course activities include Lectures, Supervised Tutorials and Practical Exercises as per teaching scheme. The programmes in C++ would be prepared during tutorials and would be executed during practicals sessions. Following needs attention:

- i. Concepts will be introduced in lectures using charts.
- ii. Role play by students for explaining concepts of array, stack, queue and sorting
- iii. Problem solving will be done through tutorials.
- iv. Practical work will be through laboratory sessions.
- v. Debate/Group Discussions for comparison of searching and sorting methods
- vi. Solving Puzzles

9. SUGGESTED LEARNING RESOURCES

A. List of Books

S. No.	Title of Books	Author	Publication
1	Data and File Structures using C	Thareja, Reema	Oxford University Press New Delhi 2011
2	Data Structures using C	ISRD Group	McGraw Hill, New Delhi, 2013
3	Data Structures	Chitra, A Rajan, P T	Tata McGraw Hill, New delhi, 2009
4	Classic Data Structures	Samanta, D.	PHI Learning, New Delhi

B. List of Major Equipment/Materials with broad specifications

- i. Hardware: Computer System with minimum PIV processor (or equivalent) and minimum 1 GB MB RAM.
- ii. Software: Turbo C++/ Borland C++/ any other higher software

C. List of Software/Learning Websites

- i. Turbo C/C++ or Borland C/C++ or any software that support c/c++ compiler
- ii. Data Structure
Introduction:<http://nptel.iitm.ac.in/video.php?subjectId=106102064>
- iii. Data Structure Tutorial: <http://www.roseindia.net/tutorial/datastructure/>
- iv. Data Structure Fundamentals: <http://www.cprograms.in/index.htm>
- v. Data structure video<http://www.youtube.com/watch?v=tORLeHHtazM>

10. COURSE CURRICULUM DEVELOPMENT COMMITTEE**Faculty Members from Polytechnics**

- **Dr. P.P.Kotak**, Head Deptt. of Computer Engineering, AVPTI, Rajkot
- **Prof. K. N. Raval**, Head Deptt. of Computer Engg., RCTI, Ahmedabad
- **Prof. Rahul B. Pancholi**, Lecturer, Deptt. of Computer Engg., L. J. Ahmedabad
- **Prof.(Mrs.) R.K.Vaghela**, Deptt. of Computer Engg., RCTI , Ahmedabad

Coordinator and Faculty Members from NITTTR Bhopal

- **Dr. Shailendra Singh**, Professor & Head Deptt. of Computer Engg. and Applications
- **Dr. PriyankaTripathi**, Associate Professor, Deptt. of Computer Engg. and Applications