Course Syllabus for CSE-242

1. Title: Data Structure (Sessional)

2. Credits: 1.5 (3 hours of lab work per week) **Session:** 2019-20

3. Course Teacher: Lamia Alam, Assistant Professor, Dept. of CSE, CUET
Md. Sabir Hossain, Assistant Professor, Dept. of CSE, CUET
Md. Billal Hossain, Lecturer, Dept. of CSE, CUET

4. Learning Resources:

Textbook(s): Schaum's Outline of Data Structures

Author: Seymour Lipschutz, **Publisher:** McGraw-Hill Education

Reference:

5. Catalog Description: Sessional based on the following topics:

Concepts and examples of elementary data objects, elementary data structures, array, stacks and queues. Lists, Trees, Graphs, heaps, B-trees, R-trees, AVL & Splay trees, Fibonacci heaps, Recursion, Memory management, Sorting and searching, hash techniques.

6. Prerequisite(s): None

7. Course Designation as Elective or Required: Required

8. Course Objectives:

a) To impart a basic understanding to identify the relative advantages and disadvantages of fundamental data structures (both linear and non-linear) and to implement them.
b) To provide the knowledge so that students can understand, implement, and calculate the time and space efficiency of classic search, sort, and traversal algorithms, including the use of big-Oh notation.
c) To enhance programming skills.

9. Student Learning Outcomes: After successfully completing the course with a grade of C (2.25/4.0) or better, the student should be able to do the following

No.	Course Learning Outcomes (CLOs)	POs#
1	Analyze a problem and identify the tradeoffs between different implementation of data structures and algorithms to make appropriate design decisions based on application data requirements to solve a problem	2
2	Design and implements fundamental data structures and algorithms to solve a variety of computational problems.	3
3	Evaluate the computational efficiency of the principal algorithms for sorting, searching, and hashing	4

10. Program Outcomes Addressed: 1, 2 and 3.

CLO#	**Program Outcome (PO)**	**PO#**
1	Problem analysis	**2**
2	Design/Development of Solution	**3**
3	Investigation	**4**

CLO—PO Mapping

No.	Course Learning Outcomes (CLOs)	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
1	Analyze a problem and identify the tradeoffs between different implementation of data structures and algorithms to make appropriate design decisions based on application data requirements to solve a problem		X										
2	Design and implements fundamental data structures and algorithms to solve a variety of computational problems.			X									
3	Evaluate the computational efficiency of the principal algorithms for sorting, searching, and hashing				X								

11. Assessment Strategy: According to the Undergraduate Academic Rule of the University

Lesson Plan

with

Lesson Learning Outcomes (LLOs)

`	Topic	Lesson Learning Outcomes (at the end of the lesson students will be able to …)	Teaching-Learning Methodology	Assessment Method
Week-01	Introduction to Data Structure	• Summarize the learning outcome of this course • Identify the difference between linear and non-linear data structure. • Recognize the importance of Data Structures in computing • State prerequisite for Data Structures implementation	• Lecture on history and theatrical background • Hands on demonstration on implementation	Lab Performance, Quiz, Viva-Voce
Week -02	Manipulation of Array	• Design and Implement List data structure using an array • Demonstrate various operations on array e.g. inserting, deleting, searching/ finding, rotating, reversing etc.	• Lecture on theatrical background • Hands on demonstration on implementation	Lab Performance, Quiz, Viva-Voce
Week -03	Applications of Array and Matrix	• Implement searching (i.e. linear and binary search) and sorting (i.e.	• Lecture on theatrical background • Hands on demonstration on implementation	Lab Performance, Quiz, Viva-Voce

		bubble sort) algorithms of using arrays • Demonstrate matrix manipulation using two-dimensional array		
Week -04	Singly Linked List	• Perform the following: a) Create a singly linked list; b) Insert a given element to the above linked list; c) Delete a given element from the above linked list; d) Display the contents of (traverse) the above list; e) Find the length of the list; f) Search an element in the list; g) Reverse the list;	• Lecture on theatrical background • Hands on demonstration on implementation	Lab Performance, Quiz, Viva-Voce
Week -05	Doubly and Circular Linked List	• Perform the following: a) Create a doubly/ circular linked list; b) Insert a given element to the above linked list; c) Delete a given element from the above linked list; d) Display the contents of (traverse) the above list; e) Find the length of the list; f) Search an element in the list; g) Reverse the list;	• Lecture on theatrical background • Hands on demonstration on implementation	Lab Performance, Quiz, Viva-Voce

Week -06	Lab Test- 1			
Week -07	Stack and Recursion	• Design and Implement stack using i) array ii) singly linked list • Use stack operations to convert a given infix expression into its postfix equivalent and evaluate the expression • Implement recursive procedures using a stack	• Lecture on theatrical background • Hands on demonstration on implementation	Lab Performance, Quiz, Viva-Voce
Week -08	Queue	• Design and Implement Queue using i) array ii) singly linked list • Design and Implement basic operations on Circular Queue	• Lecture on theatrical background • Hands on demonstration on implementation	Lab Performance, Quiz, Viva-Voce
Week -09	Sorting, Searching and Hashing	• Implement various sorting (i.e. insertion sort, shell sort, selection sort, merge sort, counting sort, and radix sort), searching (i.e. ternary search, jump search, and interpolation search) and hashing techniques	• Lecture on theatrical background • Hands on demonstration on implementation	Lab Performance, Quiz, Viva-Voce
Week -10	Trees	• Perform the following: a) Create a tree (i.e. binary tree, binary search tree, AVL tree, m-way search tree, B- tree, B+ -tree, red-black tree;) b) Insert a node to the tree; c) Delete a node from the tree;	• Lecture on theatrical background • Hands on demonstration on implementation	Lab Performance, Quiz, Viva-Voce

		d) Traverse the tree; e) Search an element in the tree; • Define a heap and implement heap sort • Implement Huffman's Algorithm		
Week -11	Graphs	• Demonstrate sequential representation of Graphs • Implement Warshall's algorithm to generate path matrix and Dijsktra's algorithm to determine shortest path in graph • Implement graph traversal algorithms: a)Depth first traversal b)Breadth first traversal	• Lecture on theatrical background • Hands on demonstration on implementation	Lab Performance, Quiz, Viva-Voce
Week-12	Lab Test- 2			
Week -13	Viva-voce and Quiz			