Description of Course CSE 113

PART A: General Information

1	**Course Title**	: Data Structure & Algorithms
2	**Type of Course**	: THEORY
3	**Offered to**	: DEPARTMENT OF IOT AND ROBOTICS ENGINEERING (IRE)
4	**Pre-requisite Course(s)**	: PROG 101

PART B: Course Details

1. Course Content (As approved by the Academic Council)

Introduction to Data Structures; Array; Algorithm Complexity; Solving Recurrences; Methods for the design of efficient algorithms; Linked List; Stack; Queue; Recursion; Tree; Graph; Sorting; Set Operations; String ADT.

2. Course Objectives

The students are expected to:

1. Allow to assess how the choice of data structures and algorithm design methods impacts the performance of programs
2. To understand the concepts of searching and sorting techniques.
3. To solve problems using data structures such as linear lists, stacks, queues, binary trees, binary search trees, and graphs and writing programs for these solutions.
4. To efficiently implement the different data structures and solutions for specific problems.

3. Knowledge required

Technical

- Introductory knowledge on Computers is required. Knowledge on any high-level programming language such as C++/Java may be an added advantage for the learners.

Mathematics

- None

4. Course Outcomes (COs)

CO No.	CO Statement	Corresponding PO(s)*	Domains and Taxonomy Level(s)**	Delivery Method(s) and Activity(-ies)	Assessment Tool(s)
1	**Understand** an algorithmic approach in designing, implementing, and debugging programs for computational problems.	PO(a), PO(b)	C2	Lectures, Tutorials, assignments, Discussions	Written exams; presentation; assignment
2	**Apply** the appropriate data structures and algorithm design techniques in problem-solving.	PO(c)	C3	Lectures, Tutorials, assignments	Written exams; presentation; assignment
3	**Evaluate** the efficiency and performance of different data structures for specific problem scenarios.	PO1(b)	C4	Lectures, Tutorials, assignments	Written exams; presentation; assignment

***Program Outcomes (POs)**

PO(a): Engineering knowledge; PO(b): Problem analysis; PO(c): Design/development of solutions; PO(d): Investigation; PO(e): Modern tool usage; PO(f): The engineer and society; PO(g): Environment and sustainability; PO(h): Ethics; PO(i): Individual work and teamwork; PO(j): Communication; PO(k): Project management and finance; PO(l): Life-long learning.

****Domains**

C-Cognitive: C1: Knowledge; C2: Comprehension; C3: Application; C4: Analysis; C5: Synthesis; C6: Evaluation
A-Affective: A1: Receiving; A2: Responding; A3: Valuing; A4: Organizing; A5: Characterizing
P-Psychomotor: P1: Perception; P2: Set; P3: Guided Response; P4: Mechanism; P5: Complex Overt Response; P6: Adaptation; P7: Organization

5. Lecture/ Activity Plan

Lec#	Lecture Topics	Teaching-learning Strategy(-ies)	Assessment Strategy(-ies)	Corresponding CO(s)
1	**Introduction:** Introduction to Data Structures, Types of Data Structures, operations on data structures, idea of abstract data type.	Lectures, Tutorials, assignments	Q/A (Class Performance), Quiz	CO2
2	**Array:** Linear Array, Representation of Linear array in memory, Traversing, Insertion, Deletion, Searching: Linear Search & Binary Search, Multidimensional Array, Pointer Array, Record.	Lectures, Tutorials, assignments	Q/A (Class Performance), Quiz	CO2, CO1
3	**Sorting:** Bubble Sort, Insertion Sort, Selection Sort. **Algorithm Complexity:** Preliminary idea of an algorithm, Asymptotic Notation, runtime complexity (Big Oh notation), the preliminary idea of data structure space complexity, Trade of Time & Space complexity.	Lectures, Tutorials, assignments	Q/A (Class Performance), Quiz	CO1
4	**Solving Recurrences:** Recursion Tree, Substitution Method, and Master Method.	Lectures, Tutorials, assignments	Q/A (Class Performance), Quiz	CO1
5	**Methods for the design of efficient algorithms:** Divide and conquer, Greedy method, Dynamic programming; Backtracking etc.	Lectures, Tutorials, assignments	Q/A (Class Performance), Quiz	CO1
6	**Linked List:** Representation of Linked Lists, Singly/doubly/circular linked lists, basic operations on linked list (insertion, deletion, traverse and searching). Header and two-way linked lists.	Lectures, Tutorials, assignments	Q/A (Class Performance), Quiz	CO2, CO3

7	**Stack:** Basic stack operations (push/pop/peek), stack-class implementation using Array and linked list, in-fix to postfix expressions conversion and evaluation, balancing parentheses using stack,	Lectures, Tutorials, assignments	Q/A (Class Performance), Quiz	CO2, CO3
8	**Queue:** basic queue operations (enqueue, dequeue), circular queue/ dequeue, queue-class implementation using array and linked list, Priority queue application- Josephus problem, palindrome checker using stack and queue, **Recursion:** Basic idea of recursion, tracing output of a recursive function, applications- merge sort, permutation, combination.	Lectures, Tutorials, assignments	Q/A (Class Performance), Quiz	CO1, CO2, CO3
9	**Tree:** General Tree: Implementation, application of general tree- file system. Binary Tree: Binary tree representation using array and pointers, traversal of Binary Tree (in-order, pre-order and postorder), BST representation, basic operations on BST (creation, insertion, deletion, querying and traversing).	Lectures, Tutorials, assignments	Q/A (Class Performance), Quiz	CO2, CO3
10	**Tree:** Huffman Tree, Huffman Coding: application- searching, sets, Heap: Min-heap, max-heap, Fibonacci-heap, applications-priority queue, Self-balancing Binary Search Tree: AVL tree (rotation, insertion).	Lectures, Tutorials, assignments	Q/A (Class Performance), Quiz	CO2, CO3
11	**Graph:** Graph representation (adjacency matrix/adjacency list), basic operations on graph (node/edge insertion and deletion), traversing a graph: breadth-first search (BFS),	Lectures, Tutorials, assignments	Q/A (Class Performance), Quiz	CO2, CO3
12	**Graph:** Depth-first search (DFS), Topological Sort, Floyd Warshall's Algorithm, Minimum Spanning Tree.	Lectures, Tutorials, assignments	Q/A (Class Performance), Quiz	CO2, CO3
13	**Sorting:** merge sort, quick sort. **Set Operations:** Set representation using bitmask, set/clear bit, querying the status of a bit, toggling bit values, LSB, application of set operations.	Lectures, Tutorials, assignments	Q/A (Class Performance), Quiz	CO1,CO2

14	**Disjoint Set:** Union, find, path compression. **String ADT:** The concatenation of two strings, the extraction of substrings, searching a string for a matching substring, parsing.	Lectures, Tutorials, assignments	Q/A (Class Performance), Quiz	CO2, CO

6. Assessment Strategy

- Class Attendance: Class attendance will be recorded in every class.
- Continuous Assessments: Continuous Assessments will be held during the semester as per the institutional ordinance.
- Mid-exam: A comprehensive mid-exam will be held at the mid of the semester as per the institutional ordinance.
- Final exam: A comprehensive Final exam will be held at the end of the semester as per the institutional ordinance.

7. Distribution of Marks

Attendance:	10 %
Continuous Assessments:	30%
Mid exam:	24%
Final Exam:	36%
Total:	100%

8. Textbook/Reference

Text Book:

- Schaum'S Outlines Data Structure by Seymour Lipschutz
- Introductions to Algorithms- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, Prentice-Hall, India

Reference Book:

- Data Structures Using C 2nd Edition by Reema Thareja
- Fundamentals of Data Structures- Horowitz E. and Sahni, S Galgotia
- Algorithms in C- Sedgewick, R (1990) (Addision Wesley)
- Data Structures and Program Design in C- Kruse/Tondo/Leung (Prentice-Hall)

Course Teacher(s):

Name:	Office/Room:	E-mail and Telephone:
Suman Saha	2nd Floor, Academic Building	suman0001@bdu.ac.bd, and 01912127578

Prepared by:
Name: Suman Saha Signature:
Date of Preparation: July 8, 2023
Date of Approval by BUGS: