Course Syllabus for CSE-244

1. **Title:** Algorithms Design and Analysis (Sessional)

2. **Credits:** 1.5 (3 hours of lab work per week) **Session:** 2018-19

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

4. **Learning Resources:**
   Textbook(s): Thomas H. Coreman et al. – Introduction to Algorithms (3rd Edition)

   Reference: ***Algorithms****, 4th Edition* by Robert Sedgewick and Kevin Wayne, Addison-Wesley Professional, 2011, ISBN 0-321-57351-X.

5. **Catalog Description:** Sessional based on the following topics:
   Various searching and sorting techniques; Efficient algorithm designing techniques: Recursion and backtracking, Divide-and-conquer paradigm, Greedy method, Dynamic programming; Analysis of algorithms: Complexity analysis; Search and traversal: Basic graph searching and traversing techniques, Shortest path problems, Topological sorting, Connected components, Spanning trees.

6. **Prerequisite(s):**
   - Programming skill (C/C++)
   - Data Structure

7. **Course Designation as Elective or Required:** Required

8. **Course Objectives:**
   (a) Getting in-depth knowledge about the analysis and design procedure of algorithms
   (b) To be able to design efficient algorithms to solve a particular problem

9. **Student Learning Outcomes:** After completing the course with a grade of D (2.0/4.0) or better, the student should be able to do the following

No	Course Learning Outcomes (CLO)	POs#
1	Understand the complexity of an algorithm	1
2	Find an efficient solution to solve a particular programming problems	2
3	Apply acquired knowledge to solve real-world problems	3
4	Design a new algorithm or improve the existing algorithm for a specific problem	3, 9

10. Program Outcomes Addressed: 1, 2, 3 and 9

CLO#	Program Outcome (PO)	PO#
1	Engineering Knowledge	1
2	Problem Analysis	2
3	Design/development of solutions	3
4	Individual and teamwork	9

CLO-PO Mapping

No	Course Learning Outcomes(CLOs)	PO1	PO2	PO3	PO4	PO5	PO6	PO07	PO08	PO09	PO10	PO11	PO12
1	Understand the complexity of an algorithm	×											
2	Find efficient solution to solve a particular programming problems		×										
3	Apply acquired knowledge to solve real-world problems			×									
4	Design a new algorithm or improve the existing algorithm for a specific problem			×						×			

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

Lesson Plan

with

Lesson Learning Outcomes (LLOs)

Week	Topic	Lesson Learning Outcomes (at the end of the lesson students will be able to)	Teaching-Learning Methodology	Assessment Method
Week - 01	Course overview (Introduction to Algorithms)	• Implement problems using basic data structure (Stack, Queue, Vector, etc) • Implement basic searching and sorting algorithms • Get an idea about efficient algorithm design procedure	• Lecture on the theoretical background • Hands-on demonstration on implementation	
Week – 02	Recursion and Backtracking	• Design recursive solution for a problem • Implement basic recursive problems (Fibonacci series, Factorial of a number) • Implement problems using backtracking (All permutation of a string, N-queen problem)	• Lecture on the theoretical background Hands-on demonstration on implementation	
Week – 03	Divide and conquer	• Understand the techniques behind dividing a problem into smaller sub-problems • Implement Merge sort, Sub-array Sum using divide and conquer approach • Analyze the complexity of divide and conquer technique based solutions	• Lecture on the theoretical background • Hands-on demonstration on implementation	

Week - 04	Research/Project proposal	• Submit a research or project proposal related to the CSE-244 course to solve real-world problems or to propose a new or improved version of existing algorithms.	• Project or research idea discussion • Guidelines to prepare project/research proposal	
Week – 05	Lab test-1 (Basic searching and sorting, Recursion and backtracking, Divide and conquer)	• Find the solution for a new problem using previously learned topics • Improve coding skill		
Week – 06	Greedy techniques	• Find out if a problem can be solved using the greedy approach • Implement activity selection and job scheduling problem	• Lecture on the theoretical background • Hands-on demonstration on implementation	
Week – 07	Dynamic programming	• Understand the basic properties to solve a problem using dynamic programming • Implement various problems using dynamic programming (Fibonacci number, 0/1 knapsack, LCS, Rod cutting, Matrix chain multiplication, etc.)	• Lecture on the theoretical background • Hands-on demonstration on implementation	
Week – 08	Lab test-2 (Greedy techniques, Dynamic programming)	• Find the solution for a new problem using previously learned topics • Improve coding skill		

Week – 09	Graph traversal	• Represent graph using adjacency matrix or adjacency list • Implement basic graph traversal techniques (BFS, DFS) • Find the shortest path on the unweighted graph using BFS	• Lecture on the theoretical background • Hands-on demonstration on implementation	
Week – 10	Shortest path, Spanning tree	• Implement various shortest path algorithms (Dijkstra, Shortest path) • Implement minimum spanning of a tree using Prim's/Kruskal's algorithm	• Lecture on the theoretical background • Hands-on demonstration on implementation	
Week – 11	Final Lab Contest	• Find the solution for a new problem using previously learned topics • Improve coding skill		
Week – 12	Research/Project Presentation	• Final project/research work presentation using Poster	• Asking the project/research contributions	
Week – 13	Quiz + Viva-voce			