<table>
<tr><td>Course Title</td><td colspan="5">Algorithms</td></tr>
<tr><td>Course Code</td><td colspan="5">CSE415</td></tr>
<tr><td>Course Type</td><td colspan="5">Compulsory</td></tr>
<tr><td>Level</td><td colspan="5">Bachelor (1st Cycle)</td></tr>
<tr><td>Year / Semester</td><td colspan="5">4th Year / 8nd Semester</td></tr>
<tr><td>Teacher’s Name</td><td colspan="5">TBA</td></tr>
<tr><td>ECTS</td><td>6</td><td>Lectures / week</td><td>3 hours / 14 weeks</td><td>Laboratories / week</td><td>None</td></tr>
<tr><td>Course Purpose and Objectives</td><td colspan="5">This course is about the design and analysis of algorithms for computational problems, and how to think clearly about analyzing correctness and running time. The objective of this course is to provide the intellectual tools needed for designing and analyzing algorithms for new problems the students may face in the future. Specific algorithms for a variety of problems will be studied, such as greedy techniques, divide-and-conquer, randomized algorithms, dynamic programming, and others, as well as general design and analysis techniques.</td></tr>
<tr><td>Learning Outcomes</td><td colspan="5">Upon successful completion of the course, students will be able to:
• Explain and use fundamental algorithms and algorithmic techniques.
• Explain the use of big-O, Omega, and Theta notation to describe the amount of work done by an algorithm, and apply them to provide tight bounds on algorithmic complexity.
• Create correctness proofs and estimate the running time of a given algorithm.
• Discuss factors other than computational efficiency that influence the choice of algorithms, such as programming time, maintainability, and the use of application-specific patterns in the input data.
• Design new algorithms for specific applications, using the algorithms and algorithmic techniques presented.</td></tr>
<tr><td>Prerequisites</td><td colspan="2">CSE400</td><td>Co-requisites</td><td colspan="2">None</td></tr>
<tr><td>Course Content</td><td colspan="5">Analysis framework:

O, Θ, Ω notations Mathematical analysis: nonrecursive and recursive algorithms. Graphs, trees and their properties. Breadth- and depth-first search in graphs, topological sort, recurrences.

Divide-and-conquer:

Multiplication of Large Integers and Strassen’s Matrix Multiplication, Closest-Pair and Convex-Hull Problems by Divide-and-Conquer</td></tr>
</table>

	Brute Force and Exhaustive Search: Selection Sort and Bubble Sort, Sequential Search and Brute-Force String Matching. Exhaustive Search, Traveling Salesman Problem, Knapsack Problem, Assignment Problem, Depth-First Search and Breadth-First Search. Sorting and Selection: Randomization, Median Finding, Quick Sort, Radix Sort, selection, Lower Bound for Sorting Greedy technique: Huffman's Codes, Minimum Spanning Tree algorithms: Kruskal's Algorithm, Prim's Algorithm, single pair Shortest Paths algorithm: Dijkstra's Algorithm Dynamic Programming: Single Source Shortest Path algorithms: Warshall's and Floyd's Algorithms, Knapsack Problem, Optimal Binary Search Trees, The Knapsack Problem and Memory Functions Amortized Analysis: Aggregate Method, Accounting Method, Potential Method, Dynamic Tables Iterative Improvement: The Simplex Method, the Maximum-Flow Problem (Ford-Fulkerson method), Maximum Matching in Bipartite Graphs, the Stable Marriage Problem Limitations of Algorithm Power: Lower-Bound Arguments, Decision Trees, P, NP, and NP-complete Problems, approximation algorithms Coping with the Limitations of Algorithm Power: Backtracking, Branch-and-Bound, Approximation Algorithms for NP-hard Problems (Vertex-cover problem, the traveling salesman problem, the set-cover problem, the vertex-coloring problem).
Teaching Methodology	Face- to- face
Bibliography	Anany V. Levitin,INTRODUCTION TO THE DESIGN AND ANALYSIS OF ALGORITHMS, Addison Wesley

	T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, INTRODUCTION TO ALGORITHMS, MIT Press S. Dasgupta, C. Papadimitriou, U. Vazirani, ALGORITHMS, McGraw-Hill Jon Kleinberg and Éva Tardos. ALGORITHM DESIGN. Addison-Wesley. R. Johnsonbaugh, M. Schaefer, ALGORITHMS, Prentice Hall
Assessment	Mid – Term Examination 25% Final Examination 40% Assignments/Lab 20% Class Participation and Attendance 10% 100%
Language	English