Future Academy
Higher Future Institute for Specialized Technological Studies

Course Specification

1- Course information:	
Course Code:	CSC262
Course Title:	Algorithms
Year/level	2nd
Academic Programs	Computer Science Program (B.Sc.)
Contact hours/ week	(Theoretical = 2hrs, Practical = 2hrs), Total = 4hrs

2- Course aims:

This course aims to provide students with a detailed introduction to different algorithm design paradigms with illustrative problems. Upon completing this course, the student should understand the solid theoretical background in analyzing and designing computer algorithms. The student should be able to use critical thinking in applying suitable classical techniques and paradigms of algorithms design to solve problems encountered either in some real problems in different domains or in advanced level courses. The student will be able to develop their own versions for a given computational task and to compare their performance (time and space complexities).

3- Intended learning outcomes of the course (ILOs):

a- Knowledge and understanding:

On successful completion of this course, the student should be able to:

a1- Recognize the basic concepts and specialist theories of algorithmic design and analysis, algorithm design paradigms, optimal algorithms, complexity theory, P and NP problems, etc.

a2- Define and know what an asymptotic analysis of algorithm performance.

a3- Identify the methods used in algorithm complexity or performance: Big-O notation, powers, and logs, growth of functions, worst-, and best-case analysis of algorithms.

a4- Determine the fundamental algorithms that are used in sorting, and searching of data structures.

a5- Present the fundamental types of algorithm design paradigms such as divide-conquer, greedy, graph, and dynamic programming.

a6- Identify some useful applications of algorithms in the area of sorting, searching, and optimization problems.

b- Intellectual skills:

On completing this course, the student should be able to:

b1- Compare algorithms complexity (the asymptotic analysis), and distinguish between the worst-, average-, and best-case analysis of algorithms

b2- Conclude the major algorithms for sorting. Analyze and categorize the performance of these algorithms and the design strategies that the algorithms embody.

b3- Distinguish the different types of algorithm paradigms and evaluate when an algorithmic design situation calls for it.

b4- Design an efficient algorithm using the most suitable design paradigm that meet specified design and performance requirements.

b5- formulate innovative design to solve a problem containing a range of commercial and industrial constraints.

c- Professional and practical skills:

At the end of this course, the student will be able to:

c1- Apply important algorithmic design paradigms and methods of analysis to identify real world problems and solve them.

c2- Solve effective computer programs to solve a variety of scientific real-world problems.

c3- Apply greedy and dynamic-programming algorithms to solve practical scientific problems

c4- Examine new algorithms or modify existing ones for new applications and reason about the efficiency of the result.

c5- Analyze a computer programming language to implement efficient algorithms designed for solving real world problems.

c6- Utilize the proposed algorithms based on the time and space quality characteristics and potential trade-offs that are offered in the given problem.

d- General and transferable skills:

On successful completion of this course, the student should be able to:

d1- Management the improved problem solving skills to basic real world situations.

d2- Working in group effectively as a member/leader of a team who may plan, design.

d3- life-long learning in enhance the ability of using a range of learning resources and to manage one's own learning.

d4- A knowledge and respect of ethics and ethical standards in relation to a major area of study.

4- Course contents

Week No.	Topics/units	Number of hours		ILO's
		Lecture hours	Practical hours	
1	Fundamentals of algorithms – understanding the problem -Types and sizes of problems, problem search space	2	2	a1, d1, d2

	and instances. Designing an algorithm and data structure.			
2	Algorithm Analysis: Best, worst and average cases, asymptotic analysis.	2	2	a2, a3, b1, b3, c2, c4, c6, d1
3	Brute Force and Exhaustive search: Selection sort and Bubble Sort – Sequential search. Exhaustive search: Knapsack Problem – Traveling Salesman Problem.	2	2	a4, b2, c1, c5, d3
4	Using recurrence relations to analyze the complexity of recursive algorithms (recursion-tree method – Iteration method- substitution method- master method)+Quiz1	2	2	a2, a3, b1, b3, 5c4, d3
5	Insertion Sort Divide and Conquer – Merge sort - Quick sort	2	2	a4, a5, b2, b4, c1, c5, d4
6	Binary Tree Traversals – Strassen's Matrix Multiplication – Large Integer Multiplication – Select K^{th} Order Element.	2	2	a3, a4, b4, c1, c2, d4
7	Midterm Exam	2	2	
8	Dynamic Programming Paradigm part(I) & Examples: Fibonacci – Longest Common Subsequence	2	2	a5, a6, b4, b5, c1, c3, d4
9	Dynamic Programming Paradigm part(II) & Examples: knapsack – Text Justification – Alternating Coin Game, Optimal Binary Search Trees	2	2	a5, a6, b4, b5, c1, c3, d4
10	Greedy Paradigm & Examples Fractional Knapsak – Activity Selection – Huffman Coding – Offline crashing	2	2	a5, b4, c1, c2, c3, d4
11	Graph Algorithms: Representation & Traversal Applications+Quiz2	2	2	a5, b4, b5, c1, c4, d4
12	Breadth-first search algorithm and Depth-first search algorithm.	2	2	a5, b4, b5, c1, c4, d4
13	Minimum Spanning Trees (MST): Kruskal's Algorithm. Single Source Shortest Path: Dijkstra's Algorithm – Speeding-up Dijkstra Introduction to NP Completeness problems	2	2	a6, b4, b5, c1, c5, d4
14	Course Review	2	2	

5- Teaching and learning methods

Methods	ILO's																			
	a1	a2	a3	a4	a5	a6	b1	b2	b3	b4	b5	c1	c2	c3	c4	c5	c6	d1	d2	d3	d4
Lectures	√	√	√	√	√	√	√	√	√						√		√		√		√
Practical sections	√									√	√	√	√	√	√	√		√			
Self-learning						√		√												√	
Assays and reviews																					
Discussion groups - projects																			√		√
Brainstorming																					
Blended-learning																					
E-learning																					

6- Teaching and learning methods for Low-achieving students

- Extra teaching hours for those who need help
- More quizzes to assess their ability for understanding the course
- Encourage the team work for those students with other advanced ones to increase their participation and understanding

7- Student assessment

Assessment method	Time	Grade weight (%)	ILOs
Course Work (Tutorial Exercise and Assignments)	**Through the semester**	5%	b5, c5, d1, d2, d3, d4
Quiz 1	**30 min, Through the lecture**	5%	a1, a2, b1, b2,
Mid-term exam	**1 hr**	10 %	a1, a2, a3, b1, b2, b3
Quiz 2	**30 min, Through the lecture**	5%	a4, a5, b4,
Practical exam	**1 hr**	15 %	b4, c1, c2, c5,
Written exam	**2 hrs**	60%	a1, a2, a3, a4, a5, a6, b1, b2, b3, b4, b4, c4, c6, d4

8-List of references

8.1. Student notebooks:

- Comprehensive instructor Notes - Slides delivered to students at the end of some lectures.

8.2. Essential textbooks:

- Thomas H. Cormen, Charles E.Leiserson, Ronald L. Rivest, "Introduction To Algorithms", 4th ed. MIT Press, Cambridge, Massachusetts, London, 2022

8.3. Recommended textbooks:

- Anany Levitin, "Introduction To The Design & Analysis of Algorithms", 3rd ed. Pearson, ISBN 13: 978-0-13-231681-1, 2012

8.4. Journals, Periodical and Reports ……..etc.

…….

8.5. Websites

- **https://www.tutorialspoint.com/design_and_analysis_of_algorithms/index.htm#:~:text=Design%20and%20Analysis%20of%20Algorithms%20covers%20the%20concepts%20of%20designing,optimal%20solution%20for%20a%20problem.**
- **https://www.tutorialspoint.com/design_and_analysis_of_algorithms/index.htm#:~:text=Design%20and%20Analysis%20of%20Algorithms%20covers%20the%20concepts%20of%20designing,optimal%20solution%20for%20a%20problem.**

Course Coordinator: *Dr. Manal Ahmed*
Head of department: *Prof. Dr. Yasser F. Ramadan*
Date of Approval: 24/7/2024