



Arab Academy for Science and Technology & Maritime Transport
College of Computing and Information Technology

University/Academy: Arab Academy for Science and Technology & Maritime Transport
Faculty/Institute: College of Computing and Information Technology
Program: B. Sc. In Computer Science

Course title	Computing Algorithms
Course code	CS 312

Form no. (11A) Knowledge and skills matrix for a course

Course content	Week study	Knowledge	Intellectual skills	Professional skills	General skills
Introduction to the design and analysis of algorithms.	1	<ul style="list-style-type: none"> • Discuss the steps of problem solving. • Understand what is the complexity of an algorithm. 	<ul style="list-style-type: none"> • Analyze algorithms 		
Measuring the asymptotic growth of functions. Lower and upper bounds	2	<ul style="list-style-type: none"> • Understand asymptotic notations. • Understand what is upper and lower bounds of an algorithm. 	<ul style="list-style-type: none"> • Analyze algorithms 		<ul style="list-style-type: none"> • Learn problem solving skills
Important problem Types	3	<ul style="list-style-type: none"> • Know the different types of problems. • Know the different problem solving techniques. 	<ul style="list-style-type: none"> • Analyze algorithms 		<ul style="list-style-type: none"> • Learn problem solving skills
Brute force	4	<ul style="list-style-type: none"> • Understand what is a Brute force. • Explain selection sort • Explain bubble sort • Explain sequential Search • Explain CP & CH by Brute force 	<ul style="list-style-type: none"> • Develop algorithms using the Brute force technique. 	<ul style="list-style-type: none"> • Implement programs for solving practical problems 	<ul style="list-style-type: none"> • Show use of computing facilities
The Greedy method	5	<ul style="list-style-type: none"> • Understand what is Greedy method, why and when to use it • Explain the Knapsack problem • Understand the minimum 	<ul style="list-style-type: none"> • Develop algorithms using Greedy technique. 	<ul style="list-style-type: none"> • Implement programs for solving practical problems 	<ul style="list-style-type: none"> • Show use of computing facilities

		spanning tree problem			
Divide and conquer	6	<ul style="list-style-type: none"> • Understand the divide and conquer technique 	<ul style="list-style-type: none"> • Compare different algorithms 	<ul style="list-style-type: none"> • Implement programs for solving practical problems 	<ul style="list-style-type: none"> • Show use of computing facilities
7th week Exam	7				<ul style="list-style-type: none"> • Demonstrate the ability to make use of a range of learning resources and to manage one's own learning.
Dynamic programming I	8	<ul style="list-style-type: none"> • Understand the importance of dynamic programming • Review Dijkstra's algorithm • Explain optimal binary search tree 	<ul style="list-style-type: none"> • Develop methods for solving different DP problems 	<ul style="list-style-type: none"> • Implement programs for solving practical problems 	<ul style="list-style-type: none"> • Show use of computing facilities
Dynamic programming II	9	<ul style="list-style-type: none"> • Explain the Knapsack problem • Explain string matching 		<ul style="list-style-type: none"> • Implement programs for solving practical problems 	<ul style="list-style-type: none"> • Show use of computing facilities
Graph algorithms	10	<ul style="list-style-type: none"> • Explain closest pair algorithm • Explain Convex Hull • Explain Depth-first search, strongly connected components • Explain Breadth-first search, Dijkstra's algorithm 	<ul style="list-style-type: none"> • Analyze algorithms 	<ul style="list-style-type: none"> • Implement programs for solving practical problems 	<ul style="list-style-type: none"> • Show use of computing facilities
Computational geometry problems	11	<ul style="list-style-type: none"> • Explain Convex Hull • Explain Line segment intersection 	<ul style="list-style-type: none"> • Analyze algorithms 	<ul style="list-style-type: none"> • Implement programs for solving practical problems 	<ul style="list-style-type: none"> • Show use of computing facilities

		<ul style="list-style-type: none"> • Explain Polygon triangulation 			
12th week Exam	12				Demonstrate the ability to make use of a range of learning resources and to manage one's own learning.
Backtracking algorithms	13	<ul style="list-style-type: none"> • Understand the backtracking algorithms 	<ul style="list-style-type: none"> • Analyze algorithms 	<ul style="list-style-type: none"> • Implement programs for solving practical problems 	<ul style="list-style-type: none"> • Show use of computing facilities
Parallel Algorithms I	14	<ul style="list-style-type: none"> • Know the need for parallel computers • Learn models of computation • 	<ul style="list-style-type: none"> • Solve a problem on a parallel machine or distributed system • Analyze parallel algorithms 	<ul style="list-style-type: none"> • Implement parallel algorithms 	<ul style="list-style-type: none"> • Show use of computing facilities
Parallel Algorithms II	15	<ul style="list-style-type: none"> • Understand algorithms for searching a sorted sequence (EREW, CREW, CRCW) • Understand algorithms for searching a random sequence (EREW, CREW, CRCW, Tree, Mesh) 	<ul style="list-style-type: none"> • Analyze parallel algorithms 	<ul style="list-style-type: none"> • Implement parallel algorithms 	<ul style="list-style-type: none"> • Show use of computing facilities

Course Instructor:

Head of Department: