Course Specification

(Bachelor)

Course Title:	Design and Analysis of Algorithms
Course Code:	CSC 1302
Program:	Bachelor in Computer Science
Department:	Computer Science
College:	Computers and Information Technology
Institution:	University of Tabuk
Version:	1.0
Last Revision Date:	27 July 2022

Table of Contents

A. General information about the course:

1. Course Identification

1. Credit hours: (3 hours)					
2. Course type					
A.	☐University	☐College	☒ Department	☐Track	☐Others
B.	☒ Required		☐Elective		
3. Level/year at which this course is offered: (Level 5/ Year 3)					
4. Course general Description:					
The main objective of this course is to teach students methods of designing and analyzing algorithms. First we introduce students to algorithm analysis techniques mainly the time complexity of algorithms using asymptotic notations. Then we move to the designing strategies including brute force, exhaustive search, decrease and conquer, divide and conquer, transform and conquer, dynamic programming and greedy methods. Numerous well-known algorithms and frequent issues, including searching, sorting, graph and tree problems, string processing, combinatorial and numerical challenges, and problems with strings, are used to illustrate these algorithm design methodologies.					
5. Pre-requirements for this course (if any)**:**					
Data Structures and Algorithms - CSC1204					
6. Co-requisites for this course (if any)**:**					
N/A					
7. Course Main Objective(s):					

- Prove the correctness of simple algorithms
- Understand asymptotic notation
- Analyze simple iterative and recursive algorithms
- To be able to design algorithms to solve common problems, using strategies like divide and conquer, dynamic programming or greedy algorithms.
- Learn a variety of useful algorithms.

2. Teaching mode (mark all that apply)

No	Mode of Instruction	Contact Hours	Percentage
1	Traditional classroom	45	100%
2	E-learning	-	-
3	Hybrid • Traditional classroom • E-learning	-	-
4	Distance learning	-	-

3. Contact Hours (based on the academic semester)

No	Activity	Contact Hours
1.	**Lectures**	45
2.	**Laboratory/Studio**	-
3.	**Field**	-
4.	**Tutorial**	-
5.	**Others (specify)**	-
Total		45

B. Course Learning Outcomes (CLOs), Teaching Strategies and Assessment Methods

Code	Course Learning Outcomes	Code of PLOs aligned with program	Teaching Strategies	Assessment Methods
1.0	**Knowledge and understanding**			
1.1	**Recognize the trade-offs of using various data structures.**	K2	• **Lectures** • **Class discussions**	• **Exams** • **Class works** • **Assignments** • **Projects**
1.2	**Demonstrate mathematics in the solution of computing problems**	K1	• **Lectures** • **Class discussions**	• **Exams** • **Class works** • **Assignments** • **Projects**
1.3	**Identify the components of algorithmic solutions such as inputs,**	K2,K3,K4	• **Lectures** • **Class discussions**	• **Exams** • **Class works** • **Assignments** • **Projects**

Code	Course Learning Outcomes	Code of PLOs aligned with program	Teaching Strategies	Assessment Methods
	outputs, variables, types, data			
1.4	Understand the theory, algorithm and design of algorithmic solutions	K2,K3,K4	• Lectures • Class discussions	• Exams • Class works • Assignments • Projects
2.0	**Skills**			
2.1	Calculate big O notation memory usage and running times associated with the developed algorithms.	S2	• Lectures • Class discussions	• Exams • Class works • Assignments • Projects
2.2	Develop practical programming skills used in industry.	S2, S4	• Lectures • Class discussions	• Exams • Class works • Assignments • Projects
2.3	Apply appropriate strategy to analyze the problems such as problem decomposition and abstraction	S2	• Lectures • Class discussions	• Exams • Class works • Assignments • Projects
2.4	Apply an appropriate data structure such as stack, queue, binary tree, or graph required to solve a problem	S2	• Lectures • Class discussions	• Exams • Class works • Assignments • Projects
3.0	**Values, autonomy, and responsibility**			
3.1	Communicate and work (effectively, ethically, and professionally) (individually and in groups/teamwork) to accomplish all the assigned duties and projects.	V2	• Class discussions	• Projects

C. Course Content

No	List of Topics	Contact Hours
1.	Introduction: motivation, definitions, syllabus overview Introduction to algorithms	3
2.	Analysis framework: O, Θ, Ω notations, Calculating worst and average complexity- part 1	3
3.	Analysis framework: O, Θ, Ω notations, Calculating worst and average complexity- part 2	3
4.	Mathematical analysis: non-recursive and recursive algorithms	3
5.	Brute-force algorithms and Exhaustive search- part 1: Concepts, Selection sort, String matching, Polynomial Evaluation, Closest-Pair Problem	3
6.	Brute-force algorithms and Exhaustive search- part 2: Traveling salesman problem, Knapsack problem, Assignment problem	3
7.	Decrease and Conquer- part 1: Concepts, Decrease by a constant (usually by 1): insertion sort, graph traversal algorithms (DFS and BFS), topological sorting algorithms for generating permutations, subsets.	3
8.	Decrease and Conquer- part 2: Decrease by a constant factor (usually by half): binary search exponentiation by squaring, multiplication à la russe fake-coin puzzle, Josephus problem	3
9.	Decrease and Conquer- part 3: Variable-size decrease: Euclid's algorithm, Selection problem	3
10.	Divide-and-conquer: Concepts, Examples: Merge sort, Quick Sort	3
11.	Transform and Conquer- part 1: Concepts, instance simplification: Searching with presorting(Binary Search), Element Uniqueness with presorting	3
12.	Transform and Conquer- part 2: representation change: AVL, Multiway Search Trees (2-3 Trees), Heap and Heap Sort, Priority Queue, Heap Maximum Key Deletion, Heap Insertion	3
13.	Transform and Conquer- part 3: problem reduction: LCM, Counting number of paths.	3
14.	Dynamic programming: Concepts, Examples: Fibonacci Series, Longest Common Subsequence.	3
15.	Greedy Algorithms: Concepts, Minimum Spanning Tree: Prim's MST, Kruskal's MST, Single-Source Shortest Path: Dijkstra's algorithm	3
Total		45

D. Students Assessment Activities

No	Assessment Activities *	Assessment timing (in week no)	Percentage of Total Assessment Score
1.	Class Works (Class Activities, *Quizzes*, Homework and Projects)	1 – 13	30%
2.	Mid-Term Exams	6,12	30%
3.	Final Exam	17	40%

*Assessment Activities (i.e., Written test, oral test, oral presentation, group project, essay, etc.).

E. Learning Resources and Facilities

1. References and Learning Resources

Essential References	Introduction to the design and analysis of algorithms, Anany Levitin, Pearson, 3rd edition, 2011, ISBN 978-0-13-231681-1
Supportive References	1) Steven S. Skiena The Algorithm Design Manual Second Edition, Springer-Verlag London Limited 2008, ISBN: 978-1-84800-069-8 e-ISBN: 978-1-84800-070-4 2) Introduction to Algorithms, Tomas H. Cormen et al., MIT press, 3rd edition, 2009, ISBN 978-0262033848. 3) Algorithms, 4th edition by Robert Sedgewick and Kevin Wayne. Addison-Wesley Professional, 2011, ISBN 0-321-57351-X.
Electronic Materials	Saudi Digital Library (SDL) (www.sdl.edu.sa)
Other Learning Materials	Data Structures And Algorithms Using C#, Michael McMillan 2007, Cambridge University Press Pulaski Technical College; ISBN-13 978-0-521-87691-9.

2. Required Facilities and equipment

Items	Resources
facilities (Classrooms, laboratories, exhibition rooms, simulation rooms, etc.)	Classroom (25 seats)
Technology equipment (projector, smart board, software)	White board, Data show projector
Other equipment (depending on the nature of the specialty)	N/A

F. Assessment of Course Quality

Assessment Areas/Issues	Assessor	Assessment Methods
Effectiveness of Teaching	Faculty, Program Leaders, and Advisory Board	Both Direct and Indirect
	Students	Indirect
Effectiveness of Students Assessment	Faculty, Program Leaders, Advisory Board, and Independent Opinion	Both Direct and Indirect
Quality of Learning Resources	Faculty, Students, and Advisory Board	Indirect
The Extent to which CLOs have been Achieved	Faculty, Program Leaders, Advisory Board, and Independent Opinion	Direct (as in section B) and Indirect/Surveys
	Students	Indirect
Other	-	-

Assessors (Students, Faculty, Program Leaders, Peer Reviewer, Others (specify)

Assessment Methods (Direct, Indirect)

G. Specification Approval

COUNCIL /COMMITTEE	
REFERENCE NO.	
DATE	