هيئة تقويم التعليم والتدريب
Education & Training Evaluation Commission

اعتماد
NCAAA
T4
2020

Course Specifications

Course Title:	Algorithm Design and Analysis
Course Code:	424CCS-3
Program:	BSc in in Computer Science
Department:	Department of Computer Science
College:	College of Computer Science and Information Systems
Institution:	Najran University

Table of Contents

A. Course Identification

1. Credit hours:

2. Course type

a. University □ College □ Department ☑ Others □

b. Required ☑ Elective □

3. Level/year at which this course is offered: Year 4 / Level 10

4. Pre-requisites for this course (if any)**:**
212CSS-3

5. Co-requisites for this course (if any)**:**
N/A

6. Mode of Instruction (mark all that apply)

No	Mode of Instruction	Contact Hours	Percentage
1	Traditional classroom	30	100%
2	Blended		
3	E-learning		
4	Distance learning		
5	Other		

7. Contact Hours (based on academic semester)

No	Activity	Contact Hours
1	Lecture	30
2	Laboratory/Studio	
3	Tutorial	
4	Others (specify)	
	Total	30

B. Course Objectives and Learning Outcomes

1. Course Description

This course introduces various algorithm design paradigms and the basics of computational complexity analysis using different models of computations with the overview of mathematical essentials, space and time complexities, asymptotic notations. Design and analysis of algorithms covers linear programming, greedy algorithms, divide-and-conquer, backtracking, branch-and-bound, search methods, graph algorithms and introduction to NP-Completeness.

2. Course Main Objective

Upon the successful completion of this course, students will be able to:

- Describe important algorithmic problem types.
- Measure the efficiency of algorithms by evaluating the time complexity of an algorithm using the asymptotic notation (Big-O(O), Omega(**Ω**), Theta(**θ**))
- Analyze the expected performance of a particular algorithm in a particular context.
- Utilize mathematical techniques to analyze the efficiency of an algorithm and demonstrate the algorithmic correctness.
- Evaluate how to deal with problems for which no fast algorithms exist (NP Completeness).

3. Course Learning Outcomes

CLOs		Aligned PLOs
1	**Knowledge and Understanding**	
1.1	Describe important algorithmic problem types.	K_1
1.2		
1.3		
1...		
2	**Skills :**	
2.1	Measure the efficiency of algorithms by evaluating the time complexity of an algorithm using the asymptotic notation (Big-O(O), Omega(**Ω**), Theta(**θ**))	S_1
2.2	Analyze the expected performance of a particular algorithm in a particular context.	S_2
2.3	Use mathematical techniques to analyze the efficiency of an algorithm and demonstrate the algorithmic correctness	S_1 , S_4
2.4	Evaluate how to deal with problems for which no fast algorithms exist (NP Completeness).	S_2 , S_5
3	**Values:**	
3.1		
3.2		
3.3		
3...		

C. Course Content

No	List of Topics	Contact Hours
1	Fundamentals of algorithmic problem solving, important problem types and fundamental data structures	6
2	Asymptotic notations and mathematical analysis	3
3	Brute force	3
4	Divide and conquer	3
5	Dynamic Programming	3
6	Greedy Algorithms	3
7	Graph Algorithms	3
8	NP-completeness and reducibility	3
9	Coping with the Limitations of Algorithm Power: Backtracking, Branch and bound	3
Total		30

D. Teaching and Assessment

1. Alignment of Course Learning Outcomes with Teaching Strategies and Assessment Methods

Code	Course Learning Outcomes	Teaching Strategies	Assessment Methods
1.0	**Knowledge and Understanding**		

Code	Course Learning Outcomes	Teaching Strategies	Assessment Methods
1.1	Describe important algorithmic problem types.	Lectures, active learning, collaborative and cooperative learning and independent study assignments are used as teaching strategies. - Showing and delivering PPT presentation in the class. - Using white board to explain important points in more detail. - Motivating students to be active during class by asking questions regularly during lecture. - Giving students tutorial to solve examples.	Following methods are used to assess student's knowledge acquired in this course. - Class Quizzes. - Midterm exam (Each exam consists of multiple choice questions, true/false, fill in the blanks, and theoretical questions.) - Final Exam
1.2			
…			
2.0	**Skills**		
2.1	Measure the efficiency of algorithms by evaluating the time complexity of an algorithm using the asymptotic notation (Big-O(O), Omega(Ω), Theta(θ))	- Showing and delivering PPT presentation in the class. - Using white board to explain important points in more detail. - Motivating students to be active during class by asking questions regularly during lecture. - Motivating students to work in home, to search from internet, to read related reference books by giving them assignments related to algorithm design and analysis techniques. - Compose more real life examples in the	Following methods are used to assess student's skills in this course. - Class quizzes. - Assignment. - Midterm exam - Final Exam
2.2	Analyze the expected performance of a particular algorithm in a particular context.		
2.3	Use the mathematical techniques to analyze the efficiency of an algorithm and demonstrate the algorithmic correctness.		
2.4	Evaluate how to deal with problems for which no fast algorithms exist (NP Completeness).		

Code	Course Learning Outcomes	Teaching Strategies	Assessment Methods
		lecture relating to the surroundings of the students to draw attention that certainly helps them to concentrate more on the specific topic. - In some cases, pick one student who understood best a specific topic and let him describe in front of the class in his own manner. - Represent more easily understandable graphs/pictures in the class to describe certain topic and in that process use interesting words or interactive sounds to help students to improve their receptive memory. - Recall the topics of last lecture and the critical issues based on different topics, which certainly helps students to recall memory frequently and store that topic in their memory for long term. - Before start a new topic or at the end of each topic, students are given couple of minutes to imagine the real life scenarios relating to that topic including implementation, advantages, deficiencies etc. to improve their logical thinking	

Code	Course Learning Outcomes	Teaching Strategies	Assessment Methods
3.0	**Values**		
3.1			
3.2			
…			

2. Assessment Tasks for Students

#	Assessment task*	Week Due	Percentage of Total Assessment Score
1	Quiz # 01	3rd week	5%
2	Quiz # 01	7th week	5%
3	Assignment or mini project (presentation)	3th week	10%
4	Assignment	8th week	10%
5	Mid Term Exam	6th week	20%
7	Final Exam	12th or 13th week	50%

***Assessment task** (i.e., written test, oral test, oral presentation, group project, essay, etc.)

E. Student Academic Counseling and Support

Arrangements for availability of faculty and teaching staff for individual student consultations and academic advice:

During the whole semester, 10 hours/week are reserved for students to guide them, to help them, to explain them topic which is not clear to them etc.

F. Learning Resources and Facilities

1.Learning Resources

Required Textbooks	Anany Levitin, **Introduction to the Design and Analysis of Algorithms**, Third Edition Publication date 13 Jan 2022, Pearson-Addison Wesley,
Essential References Materials	- T.H. Cormen, C.H. Leiserson, R.L. Rivest and C. Stein, **Introduction to Algorithms**, Fourth Edition Publication date 05 Apr 2022, The MIT Press(ISBN 0-262-03293-7) & McGraw-Hill Book Company(ISBN 0-07-013151-1). Note: Handouts will be distributed in class, when appropriate, to cover some of the course topics.
Electronic Materials	N/A
Other Learning Materials	N/A

2. Facilities Required

Item	Resources
Accommodation (Classrooms, laboratories, demonstration rooms/labs, etc.)	Lecture Rooms with 20 seats and a whiteboard or a smart board.
Technology Resources (AV, data show, Smart Board, software, etc.)	Desktop/ Laptop computer Multimedia Projector.
Other Resources (Specify, e.g. if specific laboratory equipment is required, list requirements or attach a list)	A File cabinet to keep Class Stuff, Markers, papers and students Files, and a printer to print program screenshots.

G. Course Quality Evaluation

Evaluation Areas/Issues	Evaluators	Evaluation Methods
Collecting students' questionnaire about the faculty and teaching methods.	Students	Survey
Collecting students' suggestions to facilitate more during the class.	Students	Verbal discussion
Student's questioner once during semester about course learning outcomes.	Students	Indirect Survey
Achievement percentage of course learning outcomes, direct evaluation using CLO assessment sheet	Course Instructor	Direct evaluation using CLO achievement calculation

Evaluation areas (e.g., Effectiveness of teaching and assessment, Extent of achievement of course learning outcomes, Quality of learning resources, etc.)
Evaluators (Students, Faculty, Program Leaders, Peer Reviewer, Others (specify)
Assessment Methods (Direct, Indirect)

H. Specification Approval Data

Council / Committee	Computer Science Departmental Council
Reference No.	14440203-0185-00002
Date	1st Sep, 2022