Course Specifications

Course Title:	Computational Complexity
Course Code:	CSI 413
Program:	Computer Science and Information Technology
Department:	Department of Computer Science and Information
College:	College of Science
Institution:	Majmaah University

Table of Contents

A. Course Identification

1. Credit hours: 3 **credit hours (2 lecture + 2 Exercise)**	
2. Course type **a.** University ☐ College ☐ Department ☑ Others ☐ **b.** Required ☑ Elective ☐	
3. Level/year at which this course is offered: **7th level /3**	
4. Pre-requisites for this course (if any)**:** **Discrete Mathematics of Computer Science (2) (CSI 222)**	
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**	48	80%
2	**Blended**	6	10%
3	**E-learning**	6	10%
4	**Correspondence**	0	0%
5	**Other**	0	0%

7. Actual Learning Hours (based on academic semester)

No	Activity	Learning Hours
Contact Hours		
1	**Lecture**	30
2	**Laboratory/Studio**	0
3	**Tutorial**	30
4	**Others** (specify)	0
	Total	60
Other Learning Hours*		
1	**Study**	45
2	**Assignments**	15
3	**Library**	5
4	**Projects/Research Essays/Theses**	5
5	**Others** (specify)	0
	Total	70

* The length of time that a learner takes to complete learning activities that lead to achievement of course learning outcomes, such as study time, homework assignments, projects, preparing presentations, library times

B. Course Objectives and Learning Outcomes

1. Course Description

This course is an introduction to the theory of computational complexity and standard complexity classes. One of the most important insights to have emerged from Theoretical Computer Science is that computational problems can be classified according to how difficult they are to solve. This classification has shown that many computational problems are impossible to solve, and many more are impractical to solve in a reasonable amount of time. To classify problems in this way, one needs a rigorous model of computation, and a means of comparing problems of different kinds. This course introduces these ideas, and shows how they can be used.

2. Course Main Objective

1. To analyze computational problems from a complexity perspective, and so locate them within the complexity landscape.
2. To develop skills in conducting a completeness proof, which is in a sense a practical skill.
3. To study the topic in sufficient depth as to gain an appreciation of some of the challenging issues in computer science today (e.g., P =? NP).
4. To State precisely what it means to reduce one problem to another, and construct reductions for simple examples.

3. Course Learning Outcomes

	CLOs	Aligned PLOs
1	**Knowledge:**	
1.1	Define different models of computation including Turing machine.	
1.2	Define the classes P, NP, PSPACE, and their completeness classes.	
2	**Skills :**	
2.1	Classify decision problems into appropriate complexity classes, including P, NP, PSPACE and use this information effectively.	
2.2	Reduce one problem to another, and construct reductions for simple examples.	
2.3	Provide examples of classic NP-Complete problems.	
2.4	Prove that a problem is NP-Complete by reducing a classic known one to it.	
3	**Competence:**	
3.1	work cooperatively in a small group environment.	

C. Course Content

No	List of Topics	Contact Hours
1	**Basic Concepts:** Easy and Hard Problems. Algorithms and Complexity. Examples.	**4**
2	**Turing Machines:** Models of Computation. Turing Machine, Multitape and non-deterministic Turing Machine. Decision Problems.	8

3	**Decidability. Decidable languages, Halting problem,** Counting and diagonalization. Universal Turing machine. Undecidability of Halting. Reductions.	8
4	**Reducibility:** Undecidable problems, Reductions via computation histories, Mapping reducibility.	8
5	**Time Complexity.** Measuring Complexity. The class P. The class NP. NP-completeness. Examples of NP-complete Problems.	16
6	**Space Complexity:** Savitch's Theorem. The class PSPACE, PSPACE-completeness: TQBF problem. The class L and NL. NL-Completeness. NL equals to coNL.	16
Total		

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**		
1.1	Define different models of computation including Turing machine.	Lectures Lab demonstrations Case studies Individual presentations	Written Exam Homework assignments Lab assignments Class Activities Quizzes
1.2	Define the classes P, NP, PSPACE, and their completeness classes.		
2.0	**Skills**		
2.1	Classify decision problems into appropriate complexity classes, including P, NP, PSPACE and use this information effectively.	Lectures Lab demonstrations Case studies Individual presentations Brainstorming	Written Exam Homework assignments Lab assignments Class Activities Quizzes Observations
2.2	Reduce one problem to another, and construct reductions for simple examples.		
2.3	Provide examples of classic NP-Complete problems.		
2.4	Prove that a problem is NP-Complete by reducing a classic known one to it.		
3.0	**Competence**		
3.1	work cooperatively in a small group environment.	Small group discussion Whole group discussion Brainstorming Presentation	Observations Homework assignments Lab assignments Class Activities

2. Assessment Tasks for Students

#	Assessment task*	Week Due	Percentage of Total Assessment Score
1	First written mid-term exam	6	15%
2	Second written mid-term exam	12	15%
3	Presentation, class activities, and group discussion	Every week	10%
4	Homework assignments	After each chapter	10%
5	Implementation of presented algorithms	Every two weeks	10%
6	Final written exam	16	40%
7	total		100%
8			

***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 :

Office hours: Sun: 10-12, Mon. 10-12, Thru 8-10
Office call: Sun. 12-1 and Wed 12-1

Email: h.haly@mu.edu.sa
Mobile: 0538231332

F. Learning Resources and Facilities

1.Learning Resources

Required Textbooks	Michael Sipser, Introduction to Theory of Computation, Course Technology; 3rd edition, 2012. **ISBN-10:** 113318779X and **ISBN-13:** 978-1133187790
Essential References Materials	S. Arora and B. Barak " Computational Complexity: A modern Approach", Cambridge University Press, 2009. ISBN 978-0-521-42426-4 O. Goldreich " Computational Complexity: A Conceptual Perspective", Cambridge University Press, 2008, ISBN 978-0-521-88473-0.
Electronic Materials	Video lectures are available for students at the time of the course.
Other Learning Materials	**http://mitpress.mit.edu/catalog/item/default.asp?ttype=2&tid=3440**

2. Facilities Required

Item	Resources
Accommodation (Classrooms, laboratories, demonstration rooms/labs, etc.)	Classroom and Labe available at College of science in Zulfi.
Technology Resources (AV, data show, Smart Board, software, etc.)	All resource are available in the halls
Other Resources (Specify, e.g. if specific laboratory equipment is required, list requirements or attach a list)	N\A

G. Course Quality Evaluation

Evaluation Areas/Issues	Evaluators	Evaluation Methods
Effectiveness of teaching and assessment	Students Reviewers	Questionnaires (course evaluation) filled by the students and electronically organized by the university. Student-faculty and management meetings.
Quality of learning resources	Program Leaders	Direct/indirect

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	**Dr Fayez AlFayez** **Dr. Hassan Aly**
Reference No.	4134
Date	13-10-2019