جامعة المجمعة
Majmaah University

Course Specifications

Institution:	Majmaah University
Academic Department :	Department of Computer Science and Information
Programme :	Computer Science and Information
Course :	**Computational Complexity**
Course Coordinator :	**Assoc. Prof. Hassan Aly**
Programme Coordinator :	Assoc. Prof. Youssery Azzam
Course Specification Approved Date :	22/ 12 / 1435 H

This form compatible with NCAAA 2013 Edition

A. Course Identification and General Information

1 - Course title : **Computational Complexity**	Course Code: **CSI-413**
2. Credit hours :	3 **credit hours (2 lecture + 2 Exercise)**
3 - Program(s) in which the course is offered:	**Computer Science and Information Technology Program**
4 – Course Language :	English
5 - Name of faculty member responsible for the course:	**Dr. Hassan Aly**
6 - Level/year at which this course is offered :	**7th level – 3**
7 - Pre-requisites for this course (if any) :	• **Discrete Mathematics of Computer Science 2 (CSI 222)**
8 - Co-requisites for this course (if any) :	• **None**
9 - Location if not on main campus :	**College of Science at AzZulfi**

10 - Mode of Instruction (mark all that apply)

A - Traditional classroom	√	What percentage?	**80 %**
B - Blended (traditional and online)	√	What percentage?	**10 %**
D - e-learning	√	What percentage?	**10 %**
E - Correspondence		What percentage?	**……**
F - Other		What percentage?	**…..**

Comments :

B Objectives

What is the main purpose for this course?

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.

The purpose of this course is to

1. analyze computational problems from a complexity perspective, and so locate them within the complexity landscape.

2. develop skills in conducting a completeness proof, which is in a sense a practical skill.
3. 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. State precisely what it means to reduce one problem to another, and construct reductions for simple examples.

Briefly describe any plans for developing and improving the course that are being implemented :

1. Using group discussion through the internet with course attending students.
2. Updating the materials of the course to cover the new topics of the field.
3. Increasing the ability of the students to implement the algorithms that are presented in the course.

C. Course Description

1. Topics to be Covered

List of Topics	No. of Weeks	Contact Hours
1. Basic Concepts: Easy and Hard Problems. Algorithms and Complexity. Examples.	1	4
2. Turing Machines: Models of Computation. Turing Machine, Multitape and non-deterministic Turing Machine. Decision Problems.	2	8
3. Decidability. Decidable languages, Halting problem, Counting and diagonalization. Universal Turing machine. Undecidability of Halting. Reductions.	2	8
4. Reducibility: Undecidable problems, Reductions via computation histories, Mapping reducibility.	2	8
5. Time Complexity. Measuring Complexity. The class P. The class NP. NP-completeness. Examples of NP-complete Problems.	4	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.	4	16

2. Course components (total contact hours and credits per semester):

	Lecture	Tutorial	Laboratory	Practical	Other:	Total
Contact Hours	30	30	-	-	-	60
Credit	30	15	-	-	-	45

3. Additional private study/learning hours expected for students per week.

5

The private self-study of my student is crucial for this course. It includes:

- reading carefully the topics in the textbook or reference book,
- implementing algorithms using C++ ,
- browsing the websites that concerned with the course,
- solving the exercises that are assigned in each chapter,
- discussing the course topics with the instructor in his office hours,
- watching the video lectures of other instructors who presented related topics worldwide.

The total workload of the student in this course is then: 60 + 5 x 15 = 135 work hours.

4. Course Learning Outcomes in NQF Domains of Learning and Alignment with Assessment Methods and Teaching Strategy

	NQF Learning Domains And Course Learning Outcomes	Course Teaching Strategies	Course 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	**Cognitive 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.		
3.0	**Interpersonal Skills & Responsibility**		
3.1	Provide examples of classic NP-Complete problems.	Small group discussion Whole group discussion Brainstorming Presentation	Written Exam Homework assignments Lab assignments Class Activities Quizzes
3.2	Prove that a problem is NP-Complete by reducing a classic known one to it.		
4.0	**Communication, Information Technology, Numerical**		
4.1	work cooperatively in a small group environment.	Small group discussion Whole group discussion Brainstorming Presentation	Observations Homework assignments Lab assignments Class Activities
4.2			
5.0	**Psychomotor**		

5. Schedule of Assessment Tasks for Students During the Semester:

	Assessment task	Week Due	Proportion of Total Assessment
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%

D. Student Academic Counseling and Support

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

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

E. Learning Resources

1. List Required Textbooks :

Michael Sipser, Introduction to Theory of Computation, Course Technology; 3rd edition, 2012.
ISBN-10: 113318779X and **ISBN-13:** 978-1133187790

2. List 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.

3. List Recommended Textbooks and Reference Material :

- **Journal of Computational Complexity.**

4. List Electronic Materials :

- **http://mitpress.mit.edu/catalog/item/default.asp?ttype=2&tid=3440**

5. Other learning material :

- **Video and presentation are available with me**

F. Facilities Required

1. Accommodation

- Classroom and Lab, as those that are available at college of science at AzZulfi.

2. Computing resources

- Smart Board

3. Other resources

- None

G Course Evaluation and Improvement Processes

1 Strategies for Obtaining Student Feedback on Effectiveness of Teaching:

- Questionnaires (course evaluation) achieved by the students and it is electronically organized by the university.
- Student-faculty management meetings.

2 Other Strategies for Evaluation of Teaching by the Program/Department Instructor :

- Discussion within the staff members teaching the course
- Departmental internal review of the course.

3 Processes for Improvement of Teaching :

- Periodical departmental revision of methods of teaching.
- Monitoring of teaching activates by senior faculty members.
- Training course.

4. Processes for Verifying Standards of Student Achievement

- Reviewing the final exam questions and a sample of the answers of the students by others.
- Visiting the other institutions that introduce the same course one time per semester.
- Watching the videos of other courses by international institutions.

5 Describe the planning arrangements for periodically reviewing course effectiveness and planning for improvement :

- Course evaluation
- Exam evaluation
- Improvement plan

Course Specification Approved
Department Official Meeting No (6) Date 22 / 12 / 1435 *H*

Course's Coordinator		Department Head	
Name :	Hassan Aly	***Name :***	.....
Signature :	.....	***Signature :***	.....
Date :	22/12 /1435 *H*	***Date :***	…./ … / …… *H*