Faculty of Science and Information Technology

Department: Computer Science

COURSE SYLLABUS

Theory of Computation

Student's Copy

One copy of this course syllabus is provided to each student registered in this course. It should be kept secure and retained for future use.

1. Course Information

1.	Course Title	:	**Theory of Computation**
2.	Corse Code	:	**: 1306752**
3.	Credit Hours	:	3
4.	Prerequisite	:	None
5.	Corequisite	:	None

2. Instructor Information

1.	Instructor	:	Dr Rafat Alshorman
2.	Office	:	**211D**
3.	Phone	:	
4.	Email	:	Rafat_sh@zu.edu.jo
5.	Office Hours	:	**Sun,Tu, Thu, 10-11 AM ,Sat : 8:00-9:00 AM**

3. Class Time and Place

1.	Class Days and Time:		Sat 9-12 A M
2.	Class Location	:	334
3.	Lab Days and Time	:	------
4.	Lab Location	:	------

4. Course Policies

University regulations are applied to this course, regarding Class Attendance; Punctuality, Exam, Makeup Exams; Absence with permission; Penalties for Cheating; and Policies for Assignment and Projects. Students Should be aware of all those in addition to other rules and regulations.

5. Resources

Main Reference Text Book: **.Harry R. Lewis and Christos H. Papadimitriou, Elements of theory of computation,1998.**

Additional Reference (s):

1. **Steven Bird, Ewan Klein, and Edward Loper, Natural Language Processing with Python, Published by O'Reilly ,2009**

2- Peter Linz, An Introduction to Formal Languages and Automata,2000.

6. Course Description and Purpose

1. **Theory of Computation – 3 Credits.**
2. **Purpose: The purpose of this course is to achieving the following purposes**:

- Understand mathematical and statistical methods appropriate to computer science.
- Understand formal methods and description techniques
- The notion of a formal grammar arises from the need to formalize the informal notions of grammar and language. Many formal grammars were invented: right-linear grammars, context-free grammars and unrestricted grammars. These grammars can be placed in a natural hierarchy.
- This course will also briefly cover the impact of formal language theory for many computer science applications: in compilers, natural language processing, and program verification.
- We will concentrate on three classes of models: models with finite amount of memory (finite-state automata); models with stack memory (push-down automata); and unrestricted models (Turing machines).
- Provide students the basic concepts of pigeon hole theory.
- Provide students the basic concepts of Pumping Lemma.

- 3. **Course Description:** We will study the fundamental models of computing and understand their theoretical limitations. The course will go over three models of computation; the finite automaton, the pushdown automaton and the Turing Machine. We will study the class of problems that can be solved in each of these models.

.

7. Course Learning Outcomes

Upon successful completion of this course, the learner should be able to:

A- Knowledge and understanding (students should):-

- Explore the connection between abstract machine models and formal languages, as specified by grammars.

- Enhance students' awareness of both the power and inherent limitations of algorithmic computation via the study of Turing machines and/or other abstract computational models.

- Understand mathematical and statistical methods appropriate to computer science

- Understand formal methods and description techniques.

B- Intellectual skills with ability to:-

- Apply the basic principles computational theory in problem solving.
- Apply the computational theory techniques in building formal languages.
- Apply the computational theory techniques in formal verifications.

- Apply the computational theory techniques in model checking.

C- Subject Specific Skills:

At the end of the course, students will be able to:

- extract an abstract computational model from a real world problem
- Realize the relevant between mathematics and computer science.
- Use the concepts of automat, push-down automata and Turing Machine in solving the problems.

D- Transferable skills – with ability to:-

- Distinguish between computationally tractable an intractable problems in computer science.
- Extract an abstract computational model from a real world problem.

8. Methods Of Teaching

The methods of instruction may include, but are not limited to:

1. Lectures
2. Discussion and problem solving
3. Brainstorming
4. Individual assignments
5. Case Study
6. Asking students to give a presentation in a specific subject or problem related to the course
7. Lecturing using PowerPoint Presentations, mixed with discussion with students
8. Asking students to prepare a term paper about a subject or a problem related to the course, and discuss it in the class.

9. Course Learning Assessment/Evaluation

The following methods of learning assessment will be used in this course:

	Assessment	Weight	Description
a	2 Tests - Mid Exam - Final Exam	 30% 40%	- proofs questions - Short answers - Essay Questions - Problem solving - Explanations
c	Assignments Research proposal	20%	- Asking students to prepare a term paper about a subject or a problem related to the course, and discuss it in the class
d	Presentations/participation	10%	- Student participation - Course portfolio
	Total	100%	

Note: The details for the above methods of assessment are presented below:

(a) Tests

Test	Weight %	CLO	Due Date
Mid	30%	1-5	Week 7
Final	40%	1-12	Week 16
Total	70%	12	

(b) Assignments

Assignment	Weight	CLO	Scope & Focus	Due Date
Assignments	20%	1-3	After MID Exam	after finish every Chapter

(c) Participation

Method	Weight	CLO	Focus & scope	Due Date
Participation& Presentation	10%	**	- Student contribution and cooperation - Course portfolio	After MID Exam
Total	10%			

All CLO's will be addressed in the students' participation, depending on the class and topic under consideration

10. Course Schedule/Calendar

Wk No.	Topic	Assignments/ workshops due date	Reference in the textbook	CLO
1,2	Introduction	non	Ch1,Ch2	1
3-5	Finite Automata	non	Ch2	2
5,6	Context-free Languages	non	Ch3	3
7	**Mid Exam**	non	Ch5	5
8	▪ Proving Regular languages	Proposals for term papers	Ch1, Ch2, Ch3, and Ch5	1-4

9	Pumping Lemma	non	Ch6	6
10,11	Turing Machines	non	Ch 7	7
12,13	Model checking and Kripke structure	non	Ch8	8
14	Term papers discussions		Ch11	11
15	review		Ch12	12
16	**Final Test**	Specified later	Ch1, Ch2, Ch3, Ch5,Ch6,Ch7,Ch8,Ch11 and Ch12	1-12

Special Equipment or Supplies

1. Personal Computer and special kind of software like: NuSMV and Fizzim.