Course Specifications

Course Title:	**Finite Automata and Computability**
Course Code:	**ICS 321**
Program:	**Information and Computer Science**
Department:	**Computer Science and Information**
College:	**College of Science at Az Zulfi**
Institution:	**Al- Majmaah University**

Table of Contents

A. Course Identification

1. Credit hours: 3
2. Course type **a.** University ■ College □ Department □ Others □ **b.** Required ■ Elective □
3. Level/year at which this course is offered: 6
4. Pre-requisites for this course (if any)**:** **ICS 120 – Discrete Structures**
5. Co-requisites for this course (if any)**:** NIL

6. Mode of Instruction (mark all that apply)

No	Mode of Instruction	Contact Hours	Percentage
1	**Traditional classroom**	✔	80
2	**Blended**	✔	5
3	**E-learning**	✔	5
4	**Correspondence**	--	5
5	**Other**	✔	5

7. Actual Learning Hours (based on academic semester)

No	Activity	Learning Hours
Contact Hours		
1	**Lecture**	30
2	**Laboratory/Studio**	30
3	**Tutorial**	--
4	**Others** (specify)	--
	Total	60
Other Learning Hours*		
1	**Study**	45
2	**Assignments**	15
3	**Library**	05
4	**Projects/Research Essays/Theses**	10
5	**Others** (specify)	00
	Total	(60+75 = 135)

* 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

The course introduces some fundamental concepts in automata theory and formal languages including grammar, finite automaton, regular expression, formal language, pushdown automaton, and Turing machine. Not only do they form basic models of computation, they are also the foundation of many branches of computer science, e.g. compilers, software engineering, concurrent systems, etc. The properties of these models will be studied and various rigorous techniques for analyzing and comparing them will be discussed, by using both formalism and examples.

2. Course Main Objective

The main objective of this course is to provide a formal connection between algorithmic problem solving and the theory of languages and automata and develop them into a mathematical (and less magical) view towards algorithmic design and in general computation itself.

3. Course Learning Outcomes

	CLOs	Aligned PLOs
1	**Knowledge:**	
1.1	Describe languages using Finite Automata, Nondeterministic Finite Automata, Regular Expressions, Context Free Grammars, Pushdown Automata, and Turing Machines.	**ICS-a2**
2	**Skills :**	
2.1	Convert among equivalently powerful notations for a language, including among DFAs, NFAs, and regular expressions, and between PDAs and CFGs.	**b1**
2.2	Determine a language's place in the Chomsky hierarchy (regular, context-free, recursively enumerable).	**b1**
3	**Competence:**	
3.1	Relate between Regular Languages, Context Free Languages, Recursive Languages, and Recursive Enumerable (or Computable) Languages	**b1**
3.2	Communicate effectively with others in Computer Science field.	**c3**

C. Course Content

No	List of Topics	Contact Hours
1	Introduction: basic concepts of Computational Models.	4
2	Languages: Strings and languages, finite specification of languages	4
3	Deterministic Finite Automaton (DFA): definitions and examples, design a DFA for a language, regular languages, operations of regular languages like complement, union, intersection, kleen star, concatenation.	8
4	Non-deterministic Finite Automata (NFA) and its deference to DFA, converting NFA to DFA.	8
5	Regular expressions (REX), Converting REX to NFA.	8
6	Proving languages non-regular, via the pumping lemma or alternative means	4
7	Context-free languages 1. Push-down automata (PDAs) 2. Relationship of PDAs and context-free grammars 3. Properties of context-free languages	12
8	Turing machines, or an equivalent formal model of universal computation, Nondeterministic Turing machines, Chomsky hierarchy The Church-Turing thesis, Computability, Rice's Theorem, Examples of un computable functions, Implications of un computability.	12
Total		60

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	Students will be able to demonstrate knowledge of basic mathematical models of computation and describe how they relate to formal languages	Lectures, Lab demonstrations Case studies Individual presentations	Written Exam Homework assignments Class & lab Activities Quizzes
2.0	**Skills**		
2.1	To understand that there are limitations on what computers can do and learn examples of unsolvable problems.	Group discussions, Lab demonstrations, Brainstorming Presentations	Home works and assignments
3.0	**Competence**		
3.1	Students will apply design and development principles in the construction of software systems of varying complexity	Group discussions, Case Studies, Brainstorming Presentations	Written Exam Homework assignments Class & lab

Code	Course Learning Outcomes	Teaching Strategies	Assessment Methods
3.2	Students will function effectively as a member of a team in order to accomplish a common goal		Activities Quizzes

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	Class activities, group discussions, Presentation	Every week	5%
4	Homework + Assignments	After Every chapter	5%
5	Final Lab Exam	15	20%
6	Final written exam	16	40%

***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: ,
Email: .....@mu.edu.sa

F. Learning Resources and Facilities

1.Learning Resources

Required Textbooks	Introduction to the Theory of Computation, Cengage Learning, Third Edition, M. Sipser, Cengage Learning, 2013, ISBN-10: 1292061170, ISBN-13: 9781292061177
Essential References Materials	1. An Introduction to the Theory of Computer Science Languages and Machines. Third Edition, Thomas A. Sudkamp, Addison Wesley, 20٠٦, ISBN-10: 1587145049 • ISBN-13: 9781587145049. 2. Introduction to Automata Theory, Languages, and Computation, Third Edition, J. E. Hopcroft., R. Motwani, and J. D. Ullman, Addison Wesley, 2007.
Electronic Materials	--
Other Learning Materials	--

2. Facilities Required

Item	Resources
Accommodation (Classrooms, laboratories, demonstration	1. Classrooms with required digital aids and to support traditional method of teaching using blackboard.

Item	Resources
rooms/labs, etc.)	2. Classrooms with proper lighting and air conditioning system integrated with the sound System /audio system. 3. Classroom with smart board interface, display screen and a computer to aid the sessions
Technology Resources (AV, data show, Smart Board, software, etc.)	Smart Board with supporting software / computers with updated versions of software as required to understand the subject concepts with quality headphones.
Other Resources (Specify, e.g. if specific laboratory equipment is required, list requirements or attach a list)	NIL

G. Course Quality Evaluation

Evaluation Areas/Issues	Evaluators	Evaluation Methods
Effectiveness of Teaching	Students Classroom Observation Committee Professional Development Unit External Reviewers – accreditation committee	Formal Classroom Observation - Direct Student Surveys - Indirect
Effectiveness of Assessment	Curriculum and Test Development Unit Curriculum Committee Assessment Committee External Reviewers	Faculty Feedback - indirect Student Feedback – indirect Course Reports
Extent of Achievement of Course Learning Outcomes	Quality Assurance Unit Curriculum and Test Development Unit	Course Reports Annual Program Review

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	
Reference No.	
Date	