Course Specification

(Bachelor)

Course Title:	Computational Structures (1)
Course Code:	SE1401
Program:	BSc in Software Engineering
Department:	Software Engineering
College:	College of Computing
Institution:	Umm Al Qura University
Version:	1.0
Last Revision Date:	22/04/2025

Table of Contents

A. General information about the course:

1. Course Identification

1. Credit hours: (3)					
2. Course type					
A.	☐University	☐College	☒ Department	☐Track	☐Others
B.	☒ Required		☐Elective		
3. Level/year at which this course is offered: (1st year/ 2nd level)					
4. Course General Description:					
In this module you will learn how mathematics and logic can be used to describe and reason about computational structures and systems. The module provides an important foundation for many core topics in Software Engineering, such as algorithm analysis, and program reasoning. This course provides a foundation in discrete mathematics, covering topics such as logic, proofs, sets, induction, graph theory, relations, and finite state machines. Students will learn to analyze statements, construct rigorous proofs, and solve problems using mathematical reasoning.					
5. Pre-requirements for this course (if any)**:**					
N/A					
6. Co-requisites for this course (if any)**:**					
N/A					
7. Course Main Objective(s):					
Upon successful completion of this course, you will be able to: 1. Introduce fundamental mathematical concepts for computational systems. 2. Develop the ability to model and solve problems using discrete mathematics. 3. Establish a foundation for advanced topics in computational theory.					

2. Teaching mode (mark all that apply)

No	Mode of Instruction	Contact Hours	Percentage
1	Traditional classroom	60	100%
2	E-learning	0	0
3	Hybrid • Traditional classroom • E-learning	0	0
4	Distance learning	0	0

3. Contact Hours (based on the academic semester)

No	Activity	Contact Hours
1.	**Lectures**	30
2.	**Laboratory/Studio**	30
3.	**Field**	0
4.	**Tutorial**	0
5.	**Others (specify)**	0
Total		60

B. Course Learning Outcomes (CLOs), Teaching Strategies and Assessment Methods

Code	Course Learning Outcomes	Code of PLOs aligned with the program	Teaching Strategies	Assessment Methods
1.0	Knowledge and understanding			
1.1	Understand fundamental mathematical concepts such as logic, sets, relations, and functions.	K1	Lectures,tutorials	Quizz, Assignments and Exam
1.2	Explain the principles of Boolean algebra and its applications in digital logic.	K1	Lectures,tutorials	Quizz, Assignments and Exam
1.3	Identify the basic properties of graphs and their role in modeling real-world problems.	K1	Lectures,tutorials	Quizz, Assignments and Exam
2.0	**Skills**			
2.1	Apply propositional and predicate logic to analyze and solve computational problems.	S1	Lectures,tutorials	Quizz, Assignments and Exam
2.2	Use set theory and relations to model relationships in data.	S2	Lectures,tutorials	Quizz, Assignments and Exam

Code	Course Learning Outcomes	Code of PLOs aligned with the program	Teaching Strategies	Assessment Methods
2.3	Develop and simplify Boolean expressions for logical circuits and algorithms.	S2	Lectures,tutorials	Quizz, Assignments and Exam
2.4	Perform modular arithmetic and apply it to cryptographic problems.	S1	Lectures,tutorials	Quizz, Assignments and Exam
3.0	Values, autonomy, and responsibility			
3.1	Appreciate the role of mathematical precision and rigor in software engineering.	V1	Lectures,tutorials	Quizz, Assignments and Exam
3.2	Develop an analytical mindset to approach problem-solving systematically.	V1	Lectures,tutorials	Quizz, Assignments and Exam

C. Course Content

No	List of Topics	Contact Hours
1.	Foundations of Logic Propositional Logic: Syntax and semantics Logical connectives Truth tables and logical equivalences Predicate Logic: Quantifiers (∀, ∃) Logical implications and equivalences Applications: Translating statements into logic Logical reasoning in algorithms	**8**
2.	Sets and Relations Basics of Sets: Set operations (union, intersection, difference, complement) Venn diagrams Cartesian products Relations: Definition and types (reflexive, symmetric, transitive)	**8**

	Equivalence relations and partitions Partial orderings Applications: Representing and manipulating data relationships	
3.	Functions Definitions: Domain, codomain, and range Injective, surjective, and bijective functions Composition and inverse functions Applications: Hashing functions Cryptographic functions	**4**
4.	Boolean Algebra Boolean variables and expressions Boolean operations (AND, OR, NOT, XOR) Simplification using laws (e.g., De Morgan's Laws) Karnaugh maps and minimization of Boolean expressions Applications: Digital circuit design Logical flow control in programming	**8**
5.	Introductory Number Theory Divisibility and the Euclidean algorithm Modular arithmetic: Operations and properties Applications in cryptography (e.g., RSA algorithm basics) Prime numbers and their computational importance	**8**
6.	Introduction to Graph Theory Basic Concepts: Graphs, vertices, and edges Types of graphs (directed, undirected, weighted) Paths and cycles: Eulerian and Hamiltonian paths Applications: Modeling networks and dependencies	**8**
7.	Proof Techniques Direct proofs Proof by contradiction Proof by induction Applications: Correctness of algorithms Problem-solving in computational systems	**8**

8.	Finitate State Machine and Automata	**8**
Total		**60**

D. Students Assessment Activities

No	Assessment Activities *	Assessment timing (in week no)	Percentage of Total Assessment Score
1.	Quizzes and Assignments	2-14	15
2.	Project	2-14	15
3.	Practicals	2-14	10
4.	Mid Term	7	20
5.	Final Exam	16-17	40

*Assessment Activities (i.e., Written test, oral test, oral presentation, group project, essay, etc.).

E. Learning Resources and Facilities

1. References and Learning Resources

Essential References	• Rosen, K. H. (2019). *Discrete mathematics and its applications* (8th ed.). McGraw-Hill Education. Gersting, J. (2014). *Mathematical structures for Computer Science*. WH Freeman.
Supportive References	• Houston, K. (2009). *How to think like a mathematician: A companion to undergraduate mathematics*. Cambridge University Press.
Electronic Materials	
Other Learning Materials	

2. Required Facilities and equipment

Items	Resources
facilities (Classrooms, laboratories, exhibition rooms, simulation rooms, etc.)	Traditional Classroom
Technology equipment (projector, smart board, software)	Multimedia Projector
Other equipment (depending on the nature of the specialty)	N/A

F. Assessment of Course Quality

Assessment Areas/Issues	Assessor	Assessment Methods
Effectiveness of teaching	Students	Direct, Indirect
Effectiveness of Students' assessment	Faculty, Peer reviewer	Direct, Indirect
Quality of learning resources	Faculty, Course coordinator	Direct, Indirect
The extent to which CLOs have been achieved	Course coordinator, Program management committee	Direct
Other		

Assessors (Students, Faculty, Program Leaders, Peer Reviewers, Others (specify)

Assessment Methods (Direct, Indirect)

G. Specification Approval

COUNCIL /COMMITTEE	**SOFTWARE ENGINEERING DEPARTMENT COUNCIL**
REFERENCE NO.	**THE 17TH MEETING FOR THE ACADEMIC YEAR 1446H**
DATE	**22/04/2025**