Course Specification

(Bachelor)

Course Title:	Software Engineering
Course Code:	CSC1301
Program:	Bachelor in Computer Science
Department:	Computer Science
College:	Faculty of Computing and Information Technology
Institution:	University of Tabuk
Version:	1.0
Last Revision Date:	27 July 2022

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: (Level 5/3rd Year)

4. Course general Description:

Software engineering is a discipline that allows us to apply engineering and computer science concepts in the development and maintenance of reliable, usable, and dependable software. There are several areas to focus on within software engineering, such as design, development, testing, maintenance, and management. The course is designed to present software engineering concepts and principles in parallel with the software development life cycle. The course will begin with an introduction to software engineering, giving you a definition of this body of knowledge, as well as a discussion of the main methodologies of software engineering. Students will then learn about the Software Development Life Cycle (SDLC), major methodologies followed by software modeling using Unified Modeling Language (UML), a standardized general-purpose modeling language used to create visual models of object-oriented software.

5. Pre-requirements for this course (if any):

CSC 1201- Object Oriented Programming

6. Co-requisites for this course (if any):

None

7. Course Main Objective(s):

- **Provide the student with the understanding of the main terminologies of software processes.**
- **Gain the knowledge and the understanding of various models and methods of professional software development.**
- **Understand the processes of requirements analysis, software design and software testing.**
- **Teach the student how to analyze and compare different techniques of system modeling.**
- **Provide the student with an understanding CASE tools role in software engineering.**
- **Provide the students with the fundamentals and skills of object oriented design**
- **Teach the student how to create different UML models for a specific problem.**
- **Train the student to share ideas and work in a team effectively and independently.**
- **Present software engineering concepts in a concise manner to an audience as well as writing**

a technical report for a specific system development.

2. Teaching mode (mark all that apply)

No	Mode of Instruction	Contact Hours	Percentage
1	Traditional classroom	45	100%
2	E-learning		
3	Hybrid • Traditional classroom • E-learning		
4	Distance learning		

3. Contact Hours (based on the academic semester)

No	Activity	Contact Hours
1.	**Lectures**	45
2.	**Laboratory/Studio**	
3.	**Field**	
4.	**Tutorial**	
5.	**Others (specify)**	
Total		45

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

Code	Course Learning Outcomes	Code of PLOs aligned with program	Teaching Strategies	Assessment Methods
1.0	**Knowledge and understanding**			
1.1	Recognize an appropriate strategy to analyze the problems such as problem decomposition and abstraction.	K1	**Lectures, class discussion**	**Exams, quizzes, assignments,**
1.2	Demonstrate the components of algorithmic solutions such as inputs and outputs.	K4	**Lectures, class discussion**	**Exams, quizzes, assignments,**

Code	Course Learning Outcomes	Code of PLOs aligned with program	Teaching Strategies	Assessment Methods
1.3	Describe the components of conceptual solutions such as entities, relationships, attributes, data integrity and security requirements.	K3	**Lectures, class discussion**	**Exams, quizzes, assignments,**
1.4	Recognize appropriate models for solution development.	K2	**Lectures, class discussion**	**Exams, quizzes, assignments,**
1.5	Outline and refine algorithmic solutions using decision table, pseudo code, or flowchart, Entity-Relationship (ER) model, state transition diagram, event diagram, sequence diagram, a computer-aided design tool.	K3	**Lectures, class discussion**	**Exams, quizzes, assignments,**
1.6	Recognize and walkthrough the correctness of algorithmic solutions.	K2	**Lectures, class discussion**	**Exams, quizzes, assignments,**
2.0	**Skills**			
2.1	Perform leadership by taking responsibility for various tasks, motivating and disciplining others as needed.	S4	**Lectures, class discussion**	**Exams, quizzes, assignments,**
2.2	Demonstrate the understanding of basics from other fields so they can participate effectively on multidisciplinary projects	S1	**Lectures, class discussion**	**Exams, quizzes, assignments,**
2.3	Use ideas relating to your discipline in terms that others outside your discipline can understand.	S2	**Lectures, class discussion**	**Exams, quizzes, assignments,**
2.4	Evaluate and describe accurately social and economic tradeoffs of	S3	**Lectures, class discussion**	**Exams, quizzes, assignments,**

Code	Course Learning Outcomes	Code of PLOs aligned with program	Teaching Strategies	Assessment Methods
	computing.			
2.5	Appraise the social impact when developing a computer system.	S5	**Lectures, class discussion**	**Exams, quizzes, assignments,**
2.6	Perform leadership by taking responsibility for various tasks, motivating and disciplining others as needed.	S4	**Lectures, class discussion**	**Exams, quizzes, assignments,**
3.0	**Values, autonomy, and responsibility**			
3.1	Organize and write software engineering documents clearly.	V1	**Lectures, class discussion**	**Exams, quizzes, assignments,**
3.2	Demonstrate how each individual team member interact to produce a working product.	V2	**Lectures, class discussion**	**Exams, quizzes, assignments,**

C. Course Content

No	List of Topics	Contact Hours
1.	Introduction to software engineering (part 1)	3
2.	Introduction to software engineering (part 2)	3
3.	Software processes (part 1)	3
4.	Software processes (part 2)	3
5.	Agile software development (part 1)	3
6.	Agile software development (part 2)	3
7.	Requirements engineering (part 1)	3
8.	Requirements engineering (part 2)	3
9.	System modeling (part 1)	3
10.	System modeling (part 2)	3
11.	Architectural design (part 1)	3
12.	Architectural design (part 2)	3
13.	Design and implementation (part 1)	3
14.	Design and implementation (part 2)	3
15.	Software testing	3
Total		**45**

D. Students Assessment Activities

No	Assessment Activities *	Assessment timing (in week no)	Percentage of Total Assessment Score
1.	Assignments	3.5.7.8	20%
2.	Quizzes	4.9	10%
3.	Mid Term 1	6-7	15%
4.	Mid Term 2	11-12	15%
5.	Final Exam	16	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	Engineering Software Products: An Introduction to Modern Software Engineering . Ian Sommerville, 2019, 1st Edition , ISBN-10 : 013521064X, ISBN-13 : 978-0135210642
Supportive References	Software Engineering, Ian Sommerville, 2018, 10th Edition, Prentice Hall, ISBN- 9332582696-978-9332582699.
Electronic Materials	https://www.smartdraw.com/
Other Learning Materials	

2. Required Facilities and equipment

Items	Resources
facilities (Classrooms, laboratories, exhibition rooms, simulation rooms, etc.)	**Classroom**
Technology equipment (projector, smart board, software)	**Data show**
Other equipment (depending on the nature of the specialty)	

F. Assessment of Course Quality

Assessment Areas/Issues	Assessor	Assessment Methods
Effectiveness of Teaching	Faculty, Program Leaders, and Advisory Board	Both Direct and Indirect
	Students	Indirect
Effectiveness of Students Assessment	Faculty, Program Leaders, Advisory Board, and Independent Opinion	Both Direct and Indirect
Quality of Learning Resources	Faculty, Students, and Advisory Board	Indirect
The Extent to which CLOs have been Achieved	Faculty, Program Leaders, Advisory Board, and Independent Opinion	Direct (as in section B) and Indirect/Surveys
	Students	Indirect

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

Assessment Methods (Direct, Indirect)

G. Specification Approval

COUNCIL /COMMITTEE	

REFERENCE NO.	
DATE	