Course Specification

(Bachelor)

Course Title:	Trusted Software Development
Course Code:	SE4711
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: (3rd year/ 5th or 6th level) or (4th year/8th level)

4. Course General Description:

The "Trusted Software Development" course offers a comprehensive examination of the methodologies and practices necessary for creating secure and reliable software systems. Students will learn how to design, analyze, and implement software that maintains confidentiality, integrity, and availability, even in the face of potential threats. The curriculum includes topics such as security policies, access control mechanisms, threat modeling, and the integration of security measures throughout the software development lifecycle.

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

SEC4012 - Secure Software Development

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

N/A

7. Course Main Objective(s):

Upon successfully completing this unit, students will be able to:

1. Understand the fundamental concepts of trusted software development.
2. Develop skills relevant to trusted software development.
3. Implement and evaluate measures to ensure trusted software development.
4. Promote ethical and professional responsibility in their work.

2. Teaching mode (mark all that apply)

No	Mode of Instruction	Contact Hours	Percentage
1	Traditional classroom	45	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**	45
2.	**Laboratory/Studio**	0
3.	**Field**	0
4.	**Tutorial**	0
5.	**Others (specify)**	0
Total		45

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	Explain the principles of trusted software development, including risk assessment.	K1	Lectures, exercises	Quizzes, Assignments, exam
1.2	Examine emerging trends in trusted software development and their applications.	K2	Lectures, exercises	Quizzes, Assignments, exam
2.0	**Skills**			
2.1	Analyse security requirements and design secure software systems using appropriate models.	S1	Case studies, exercises	Assignments, projects
2.2	Apply secure coding practices and perform vulnerability assessments.	S2	Case studies, exercises	Assignments, projects
2.3	Evaluate software systems using security assurance techniques and formal methods.	S3	Case studies, exercises	Assignments, projects
3.0	**Values, autonomy, and responsibility**			
3.1	Demonstrate ethical and professional practices in software development processes.	V1	Role-playing scenarios, ethics workshops	Reflection papers

Code	Course Learning Outcomes	Code of PLOs aligned with the program	Teaching Strategies	Assessment Methods
3.2	Collaborate effectively in teams to achieve trusted software development goals.	V2	Collaborative tools , group project	Group project,

C. Course Content

No	List of Topics	Contact Hours
1.	**Introduction to Trusted Software Development** • Understanding the concept of trust in software systems. • Security fundamentals and threat modelling. • Risk assessment and management strategies.	**6**
2.	**Security Policies and Models** • Comprehensive overview of security policies and models. • Application of security policies in software development.	**3**
3.	**Access Control Mechanisms** • Design and implementation of robust access control methods. • Techniques to protect software resources effectively.	**3**
4.	**Secure Software Development Lifecycle (SDLC)** • Incorporating security requirements engineering. • Implementing secure design and architecture principles. • Adopting secure coding practices and techniques. • Conducting thorough code reviews and static analysis.	**6**
5.	**Software Assurance Techniques** • Utilizing formal methods and verification processes. • Applying dynamic analysis and penetration testing. • Employing security testing tools and frameworks. • Conducting vulnerability assessment and remediation.	**6**
6.	**Threat Modeling and Risk Assessment** • Identifying potential threats and vulnerabilities in software systems. • Assessing and managing risks to maintain software integrity.	**3**
7.	**Secure Software Design Principles** • Applying design principles to develop secure and resilient software architectures. • Ensuring system integrity and confidentiality through design.	**3**
8.	**Implementation of Security Features** • Integrating authentication, authorization, and encryption into software applications. • Ensuring resilience against potential threats through robust implementation.	**6**

9.	**Security Testing and Vulnerability Assessment** • Techniques for testing software security and identifying vulnerabilities. • Ensuring software reliability through comprehensive testing.	**6**
10.	**Emerging Trends in Trusted Software Development** • Secure development practices in cloud computing and IoT environments. • Leveraging artificial intelligence and machine learning in security. • Exploring blockchain technology to enhance software trust	**3**
Total		**45**

D. Students Assessment Activities

No	Assessment Activities *	Assessment timing (in week no)	Percentage of Total Assessment Score
1.	Quizzes	4,12	15
2.	Group Project	12	15
3.	Assignments	2-14	10
4.	Mid Term	8	20
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	• Adkins, H., Beyer, B., Blankinship, P., Lewandowski, P., Oprea, A., & Stubblefield, A. (2020). *Building secure and reliable systems: Best practices for designing, implementing, and maintaining systems*. O'Reilly Media. ISBN 978-1492083122. • Larsen, N. (2024). *Building secure software: A hands-on guide for developers*.
Supportive References	
Electronic Materials	
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)	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**