جـامعـة الـزيـتـونـــة الأردنـيــة

Al-Zaytoonah University of Jordan

كلية العلوم وتكنولوجيا المعلومات

Faculty of Science and Information Technology

" عراقة وجودة"

"Tradition and Quality"

QF01/0408-4.0E	**Course Plan for Bachelor program - Study Plan Development and Updating Procedures/ Cyber Security Department**

Study plan No.	**2024/2025**		**University Specialization**		**Cybersecurity**	
Course No.	**0133204**		**Course name**		**Software Development Life Cycle**	
Credit Hours	**3**		**Prerequisite Co-requisite**			
Course type	□ MANDATORY UNIVERSITY REQUIREMENT	□ UNIVERSITY ELECTIVE REQUIREMENTS	□ FACULTY MANDATORY REQUIREMENT	□ ✓ Support course family requirements	□ ✓ **Mandatory requirements**	□ Elective requirements
Teaching style	□ **Full online learning**		□ ✓Blended learning		□ Traditional learning	
Teaching model	□ 2Synchronous: 1asynchronous		□ 2 face to face : 1synchronous		□ ✓3 Traditional	

Faculty member and study divisions information (to be filled in each semester by the subject instructor)

Name	**Academic rank**	**Office No.**	**Phone No.**	**E-mail**	
Dr. Mahmoud Farfoura	**Assistant professor**	**118**		**m.farfoura@zuj.edu.jo**	
Division number	**Time**	**Place**	**Number of students**	**Teaching style**	**Approved model**

Brief description

The "Software Development Life Cycle" (SDLC) course covers software development methods and processes in detail. Students will study SDLC planning, analysis, design, implementation, testing, deployment, and maintenance. Software development is iterative and user requirements are stressed in the training. Students will learn project management and quality assurance methodologies using real-world case studies and best practices. The course also covers Agile and DevOps approaches to prepare students for modern software development difficulties. Software engineering occupations require practical skills learnt through hands-on projects and teamwork.

Learning resources

Course book information (Title, author, date of issue, publisher ... etc)	Software Engineering: A Practitioner's Approach by Roger S. Pressman, 9th Edition			
Supportive learning resources (Books, databases, periodicals, software, applications, others)	The Pragmatic Programmer: Your Journey To Mastery, Andrew Hunt and David Thomas, 2019			
Supporting websites	https://www.catonetworks.com/network-security/network-security-protocols/			
The physical environment for teaching	□ **Class room**	□ ✓labs	□ **Virtual educational platform**	□ Others

Necessary equipment and software	**MS office, MS Visio.**
Supporting people with special needs	
For technical support	**E-learning and Open Educational Center. Computer Center**

Course learning outcomes (S= *Skills, C= Competences K= Knowledge,)*

No.	Course learning outcomes	The associated program learning output code
	Knowledge	
K1	Define the key phases of the Software Development Life Cycle.	**PLO1**
K2	Describe various software development methodologies, including Agile and Waterfall.	**PLO2**
K3	Explain the importance of requirement gathering and analysis in software projects.	**PLO2**
	Skills	
S1	Develop software project plans and timelines using SDLC principles.	**PLO1**
S2	Conduct testing and quality assurance measures to ensure software reliability.	**PLO3**
	Competences	
C1	Collaborate effectively in teams to complete software development projects.	**PLO2**
C2	Adapt to changing project requirements and methodologies in dynamic environments.	**PLO3**

Mechanisms for direct evaluation of learning outcomes

Type of assessment / learning style	Fully electronic learning	Blended learning	Traditional Learning (Theory Learning)	Traditional Learning (Practical Learning)
First exam	0	**0**	0	0
Second / midterm exam	%30	**%30**	%30	%30
Participation / practical applications	0	**0**	0	0
Asynchronous interactive activities	%30	**%30**	%30	%30
final exam	%40	**%40**	%40	%40

Note: Asynchronous interactive activities are activities, tasks, projects, assignments, research, studies, projects, work within student groups ... etc, which the student carries out on his own, through the virtual platform without a direct encounter with the subject teacher.

Schedule of simultaneous / face-to-face encounters and their topics

Week	Subject	learning style*	Reference **

QF01/0408-4.0E	**Course Plan for Bachelor program - Study Plan Development and Updating Procedures/ Cyber Security Department**

1	Introduction to Software Engineering	Lecture Notes	
2	Software Development Life Cycle Models	Lecture Notes	
3	Software Development Methodologies	Lecture Notes	
4	System Design and Architecture	Lecture Notes	
5	Implementation and Coding Standards	Lecture Notes	
6	Software Testing Fundamentals	Lecture Notes	
7	Testing Techniques and Tools	Lecture Notes	
8	**Midterm Exam**		
9	Quality Assurance in Software Development	Lecture Notes	
10	Deployment Strategies	Lecture Notes	
11	Maintenance and Support	Lecture Notes	
12	Project Management in Software Development	Lecture Notes	
13	Case Studies in Software Development	Lecture Notes	
14	Future Trends in Software Development	Lecture Notes	
15	Projects Discussion	Presentation and Discussion	
16	Final Exam		

*** Learning styles: Lecture, flipped learning, learning through projects, learning through problem solving, participatory learning ... etc.**

**** Reference: Pages in a book, database, recorded lecture, content on the e-learning platform, video, website ... etc.**

Schedule of asynchronous interactive activities (in the case of e-learning and blended learning)

This activities was designed using the **Project-Based Learning (PBL)**
Project 1: Software Development Lifecycle, Design & Implementation (PBL)

Task / Activity	Reference	Expected Results
Define system requirements and select methodology (Agile/Waterfall)	Software Development Methodologies	Clear SDLC model selection & justified requirements
Design system architecture, modules, and data flow	System Design and Architecture	Structured design diagrams & architecture plan
Apply coding standards during implementation	Implementation and Coding Standards	Clean, readable, maintainable code
Develop core system functionality	SDLC Implementation Phase	Functional working software prototype
Apply unit testing and basic testing concepts	Software Testing Fundamentals	Verified components functionality
Use testing techniques & basic testing tools	Testing Techniques and Tools	Defect identification and documented results
Document system design, implementation & testing	PBL Documentation Requirements	Complete structured technical report