MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information					
معلومات المادة الدراسية					

Module Title	Software Engineering		**Module Delivery**	
Module Type	Core		☒ **Theory** ☒ **Lecture** ☒ **Lab** ☐ **Tutorial** ☒ **Practical** ☐ **Seminar**	
Module Code				
ECTS Credits	6			
SWL (hr/sem)	150			
Module Level	2	**Semester of Delivery**	1	
Administering Department	CS	**College**	Computer Science	
Module Leader	م.م اسيل جاسم محمد Ass.Lec Aseel Jasim	**e-mail**		
Module Leader's Acad. Title	Assistant Lecturer	**Module Leader's Qualification**	Master Dgree	
Module Tutor		**e-mail**	E-mail	
Peer Reviewer Name	Name	**e-mail**	E-mail	
Scientific Committee Approval Date	01/06/2023	**Version Number**	1.0	

Relation with other Modules			
العلاقة مع المواد الدراسية الأخرى			

Prerequisite module		**Semester**	2
Co-requisites module	None	**Semester**	

	Module Aims, Learning Outcomes and Indicative Contents
	أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية
Module Objectives أهداف المادة الدراسية	1. Introducing students to the basic concepts of software engineering and their importance in building efficient and maintainable systems. 2. Enabling students to analyze system requirements and transform them into a well-thought-out software design using appropriate modeling tools. 3. Teaching students advanced software design principles (such as SOLID and design patterns) to create high-quality software. 4. Training students to use testing and version control tools to ensure software reliability and ease of development. 5. Providing students with software project management skills using modern methodologies such as Agile and Scrum. 6. Developing students' ability to work within a development team and understanding the ethical and professional aspects of the field. 7. Enhancing students' communication skills to document software and deliver technical presentations effectively.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	1- Understand the Software Life Cycle (SDLC) and its development models (e.g., Waterfall, Agile). 2- Analyze system requirements and design solutions using modeling tools (e.g., UML). 3- Apply software design principles (e.g., SOLID, design patterns) to write maintainable code. 4- Use testing and version control tools (e.g., JUnit, Git) to ensure software quality. 5- Manage software projects using methodologies such as Scrum and evaluate technical trade-offs. 6- Work effectively within development teams while adhering to professional ethics (e.g., security, intellectual property). 1. Document software and clearly deliver technical presentations to stakeholders and participating teams.
Indicative Contents المحتويات الإرشادية	• Introduction to Software Engineering • Engineering Systems • Engineering Systems • Processing • Management • Modeling Languages

Learning and Teaching Strategies
استراتيجيات التعلم والتعليم
Strategies	1. Lectures and interactive discussions 2. Practical laboratory sessions 3. Problem-solving exercises and tutorials 4. Simulation tools and software 5. Assessments (exams, projects) with feedback