MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information معلومات المادة الدراسية					
Module Title	Software Engineering (I)			Module Delivery	
Module Type	Core			☒ Theory ☐ Lecture ☒ Lab ☒ Tutorial ☐ Practical ☐ Seminar	
Module Code	SOFT2305				
ECTS Credits	7				
SWL (hr/sem)	175				
Module Level	2		Semester of Delivery	5	
Administering Department	SOFT		College	CIT	
Module Leader	Muyassar Dalli Hamad		e-mail	muyassar.dalli@uofninevah.edu.iq	
Module Leader's Acad. Title	Lecturer		Module Leader's Qualification	Ph.D.	
Module Tutor	Name (if available)		e-mail	E-mail	
Peer Reviewer Name	Name		e-mail	E-mail	
Scientific Committee Approval Date	20/06/2023		Version Number	1.0	

Relation with other Modules العلاقة مع المواد الدراسية الأخرى			
Prerequisite module	None	Semester	
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. Understand the basic concepts of software engineering and its role in the software development life cycle. 2. Demonstrate knowledge of software development methodologies and their use in different development scenarios. 3. Apply software engineering principles to develop reliable, maintainable, and scalable software systems. 4. Understand software design patterns and their use in solving common design problems. 5. Analyze requirements, design and develop software applications using object-oriented design principles. 6. Understand principles of software testing, debugging, and maintenance. 7. Apply appropriate tools and techniques for version control and configuration management. 8. Understand the importance of teamwork, communication, and project management skills in a software development project. 9. Analyze ethical and social issues related to software engineering practices. 10. Apply principles of software engineering to develop a significant project in a team environment.
Module Learning Outcomes **مخرجات التعلم للمادة الدراسية**	1. Describe key software engineering concepts and principles. 2. Explain the importance of software quality, reliability, and maintainability. 3. Compare and contrast different software development methodologies. 4. Apply software engineering principles to design software systems. 5. Develop software using object-oriented design principles. 6. Identify common software design patterns and their use in solving design problems. 7. Use appropriate tools and techniques for software version control and configuration management. 8. Implement software testing techniques and strategies to ensure software quality. 9. Apply debugging techniques to identify and resolve software defects. 10. Demonstrate the ability to work collaboratively in a software development team. 11. Apply software engineering principles to develop a significant software project. 12. Analyze ethical and social issues related to software engineering practices. 13. Demonstrate effective communication skills in presenting software engineering concepts and solutions. 14. Use critical thinking skills to evaluate and select appropriate software engineering solutions.
Indicative Contents **المحتويات الإرشادية**	1. Introduction to Software Engineering (2 hours) - Definition of software engineering - Software development concepts and process models - Importance of software quality 2. Software Requirements (8 hours) - Specification and analysis of software requirements - Requirement engineering process - Requirement validation and management 3. Software Design (16 hours) - Fundamentals of software design

	- Object-oriented design principles and patterns - Design validation and verification 4. Software Development Methodologies (16 hours) - Overview of various software development methodologies - Waterfall model, Agile model, and Iterative model - Hybrid models and selection criteria for different methodologies 5. Software Testing (16 hours) - Objectives and principles of software testing - Testing techniques and strategies - Testing process and testing tools 6. Software Maintenance (8 hours) - Software maintenance process and activities - Techniques for software maintenance - Software reengineering and reverse engineering 7. Software Configuration Management (8 hours) - Configuration management process and activities - Tools and techniques for configuration management - Version control and change control 8. Software Engineering Ethics and Professionalism (4 hours) - Ethical and professional issues in software engineering - Professional bodies, standards and codes of ethics 9. Software Project Management (8 hours) - Project management concepts and principles - Project planning and scheduling - Risk management 10. Software Quality Assurance (8 hours) - Quality concepts and principles - Quality assurance activities - Metrics and measures 11. Software Engineering Case Studies (8 hours) - Case studies related to software engineering problems and solutions.

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	1. Attend lectures and actively participate in class discussions to clarify the course contents and resolve any issues. 2. Take notes during lectures and review them later to reinforce the learning. 3. Read textbooks and course materials before and after lectures to enhance comprehension. 4. Do assignments and projects regularly, as they provide an opportunity to apply knowledge and skills and receive feedback. 5. Collaborate with classmates and seek their help in understanding difficult concepts. 6. Utilize online resources such as video tutorials, interactive quizzes, and forums to supplement the course material. 7. Attend workshops, seminars, and conferences related to software engineering

	to gain practical knowledge and perspective. 8. Seek mentorship from experienced software engineers or professors to gain insights and guidance. 9. Practice programming, testing, and debugging skills on various software tools and platforms. 10. Monitor your progress through self-assessment and seek feedback from instructors and peers to improve your weak areas.

Student Workload (SWL) الحمل الدراسي للطالب محسوب لـ ١٥ أسبوعا			
Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	78	Structured SWL (h/w) الحمل الدراسي المنتظم للطالب أسبوعيا	5
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	97	Unstructured SWL (h/w) الحمل الدراسي غير المنتظم للطالب أسبوعيا	7
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	175		

Module Evaluation تقييم المادة الدراسية					
As		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	Quizzes	1	10% (10)	5	LO #1, #2 , #3, #4
	Assignments	2	10% (10)	2 and 12	LO #3, #4 and #6, #7
	Projects / Lab.	2	20%(20)	6 and 11	LO #3, #4 and #8, #9
	Report				
Summative assessment	Midterm Exam	2hr	10% (10)	7	LO #1 - #7
	Final Exam	3hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus) المنهاج الاسبوعي النظري	
Week	**Material Covered**
Week 1	Introduction to Software Engineering and its history, software process models
Week 2	Software Requirement Analysis and Specification, Functional and non-functional requirements
Week 3	Software Design, Architectural patterns, design patterns
Week 4	4: Software Design, Object-oriented design principles, UML diagrams
Week 5	Software Testing, Levels of Testing, Black box and White box Testing
Week 6	Software Testing, Unit Testing, Integration Testing
Week 7	Software Configuration Management, Version Control, Continuous Integration
Week 8	Software Maintenance, Reverse Engineering, Re-engineering
Week 9	Software Project Management, Software Cost Estimation, Quality Assurance
Week 10	Agile Software Development, Scrum Methodology
Week 11	Software Engineering Ethics, Intellectual Property Rights
Week 12	Software Security and Cybersecurity, Encryption, Decryption, Cryptography
Week 13	Software Quality, Quality Models, ISO 9000 Standards
Week 14	Software Metrics, Code Metrics, Testing Metrics
Week 15	Emerging Trends in Software Engineering, Machine Learning in Software Engineering, AI and Robotics.
Week 16	**Preparatory week before the final Exam**

Delivery Plan (Weekly Lab. Syllabus) المنهاج الاسبوعي للمختبر	
Week	**Material Covered**
Week 1	ntroduction to the Lab Environment

Week 2	MIPS Assembly Language Programming
Week 3	CPU and Memory Simulation
Week 4	Pipelining Simulation
Week 5	Cache Memory Simulation
Week 6	Virtual Memory Simulation
Week 7	Input-Output Simulation
Week 8	Advanced Topics in Computer Architecture

Learning and Teaching Resources

مصادر التعلم والتدريس

	Text	Available in the Library?
Required Texts	1. Computer Organization and Design: The Hardware/Software Interface by David A. Patterson and John L. Hennessy 2. Structured Computer Organization by Andrew S. Tanenbaum 3. Computer Architecture: A Quantitative Approach by John L. Hennessy and David A. Patterson	
Recommended Texts	1 Digital Design and Computer Architecture by David Money Harris and Sarah L. Harris 2. Computer Systems: A Programmer's Perspective by Randal E. Bryant and David R. O'Hallaron 3. Operating System Concepts by Abraham Silberschatz, Greg Gagne, and Peter B. Galvin	
Websites	Internet , youTube	

Grading Scheme

مخطط الدرجات

Group	Grade	التقدير	Marks %	Definition
Success Group (50 - 100)	**A** - Excellent	**امتياز**	90 - 100	Outstanding Performance
	B - Very Good	**جيد جدا**	80 - 89	Above average with some errors
	C - Good	**جيد**	70 - 79	Sound work with notable errors

	D - Satisfactory	**متوسط**	60 - 69	Fair but with major shortcomings
	E - Sufficient	**مقبول**	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX –** Fail	**راسب (قيد المعالجة)**	(45-49)	More work required but credit awarded
	F – Fail	**راسب**	(0-44)	Considerable amount of work required

Note: Marks Decimal places above or below 0.5 will be rounded to the higher or lower full mark (for example a mark of 54.5 will be rounded to 55, whereas a mark of 54.4 will be rounded to 54. The University has a policy NOT to condone "near-pass fails" so the only adjustment to marks awarded by the original marker(s) will be the automatic rounding outlined above.