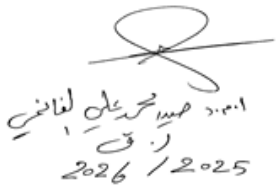
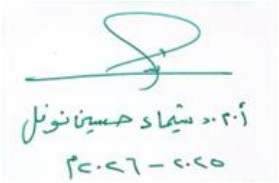
نموذج وصف المقرر الدراسي

معلومات المقرر الدراسية				
اسم المقرر	هندسة البرمجيات		أسلوب التدريس	
نوع المقرر	رئيسة		☒ محاضرة ☒ عملي	
رمز المقرر	IT3205			
عدد الوحدات	6			
عدد ساعات المقرر	150			
مستوى المقرر الدراسي	3	الفصل الدراسي	2	
القسم الأكاديمي	تكنولوجيا المعلومات	الكلية	كلية العلوم	
مسؤول المادة	م.م علي محمود علي	الايميل	ali.mahmoud@uowa.edu.iq	
اللقب العلمي	مدرس مساعد	الشهادة الاكاديمية	ماجستير	
مدرس المادة	م.م علي محمود علي	الايميل	ali.mahmoud@uowa.edu.iq	
اسم مراجع المقرر الدراسي	م.د محمود جاسم خلصان	الايميل	mahmood.jasim@uowa.edu.iq	
تاريخ موافقة اللجنة العلمية	2025-2026	اصدار	V1	

العلاقة مع المقررات الدراسية الاخرى			
المتطلب السابق للمادة	-	الفصل الدراسي	-
المتطلبات المصاحبة للمادة	-	الفصل الدراسي	-





مصادقة رئيس القسم

مصادقة السيد عميد الكلية المحترم

أهداف المادة، ومخرجات التعلم، والمحتوى الإرشادي	
هدف المادة الدراسية	يهدف هذا المقرر إلى تمكين الطلبة من تطبيق المبادئ الهندسية في تطوير البرمجيات من أجل تحسين الجودة، وتقليل الوقت، وزيادة كفاءة التكلفة، مع ضمان تنفيذ عمليات اختبار منظمة ومنهجية.
مخرجات تعلم المادة الدراسية	✓ تمكين الطالب من التعرّف على أساسيات بناء البرامج. ✓ تمكين الطالب من معرفة وفهم كيفية التعامل مع الأنواع المختلفة لأساليب تحليل البرامج. ✓ تمكين الطالب من تعلّم كيفية تصميم البرامج بصورة منهجية. ✓ تعريف الطلبة بأساسيات بناء برنامج باستخدام لغة برمجة معيّنة تُعد نموذجًا يمكن القياس عليه لبقية لغات البرمجة. ✓ تمكين الطالب من تقييم جودة البرامج المصمَّمة والتحقق من صحتها. ✓ إكساب الطالب مهارات عملية في تطبيق الأساليب المستخدمة في هندسة البرمجيات. ✓ تزويد الطالب بالمهارات اللازمة للتعامل مع أي فكرة تطبيقية وبناء مشروع متكامل. ✓ تزويد الطالب بمهارات تحديث البرامج وتصحيح الأخطاء البرمجية التي قد تظهر أثناء التنفيذ. ✓ تنمية مهارات التفكير المنظّم والتحليل لموضوع معيّن وتحويله إلى حل برمجي متكامل.
المحتوى الإرشادي	يتضمن المحتوى الارشادية ما يلي: 1. تحليل المتطلبات ونمذجتها.(Requirements Analysis and Modeling) 2. تطوير التصاميم ونمذجتها وتقييمها.(Developing, Modeling, and Evaluating Designs) 3. النمذجة باستخدام لغة النمذجة الموحدة.(Unified Modeling Language – UML) 4. عمليات تصميم البرمجيات ومبادئها.(Software Design Processes and Principles) 5. أنماط التصميم الشائعة (Design Patterns) وهندسة البرمجيات (Software Architectures). 6. الأدوات المستخدمة في التصميم والتطوير.(Tools for Design and Development)

استراتيجيات التعليم والتعلم	
استراتيجيات	✓ تتضمن استراتيجيات التعليم والتعلّم لمادة هندسة البرمجيات في قسم تقنية المعلومات منهجًا متوازنًا يجمع بين الفهم النظري والتطبيق العملي. ✓ توفر المحاضرات والمناقشات التفاعلية ودراسات الحالة الأساس النظري اللازم. ✓ تتيح التمارين العملية، والعمل الجماعي، والمشاريع اكتساب خبرة عملية مباشرة في مبادئ هندسة البرمجيات. ✓ تقدم الورش العملية، والعروض التوضيحية، وأمثلة من الصناعة رؤية واقعية للتطبيق العملي. ✓ تساعد الموارد الإلكترونية، والتقييمات، والتغذية الراجعة في تعزيز التعلم وترسيخ المفاهيم. ✓ تركز المختبرات الافتراضية والتعلّم المستمر على تطوير المهارات العملية ومواكبة أحدث الاتجاهات في الصناعة. ✓ تضمن هذه الاستراتيجيات تحقيق فهم شامل لهندسة البرمجيات وأهميتها في مجال تقنية المعلومات.

حمل عمل الطالب			
الساعات المجدولة (ساعات/فصل دراسي)	60	الساعات المجدولة (ساعات/أسبوع)	4
الساعات غير المجدولة (ساعات/فصل دراسي)	87	الساعات غير مجدولة (ساعات/أسبوع)	6
الإجمالي (ساعات/فصل دراسي)	147 + 3 نهائي = 150		

تقييم المقرر الدراسي					
		الوقت/العدد	الوزن (الدرجات)	الأسابيع	مخرجات التعلم
التقويم التكويني	اختبارات	5	10% (10)	2,4,6,8,11,14	#1, #2, #4
	واجبات داخل كلية	2	10% (10)	3,12	#2, #3, #9
	مختبر/مشروع	1	10% (10)	جميع الاسابيع	#6, #7, #8
	تقرير	1	10% (10)	13	#5, #8
التقييم النهائي	امتحان المد	2hr	10% (10)	7	#1 –#6
	امتحان النهائي	3hr	50% (50)	16	كل المخرجات
إجمالي التقييم			100% (100)		

خطة التدريس (المنهج الأسبوعي)	
	المنهج الدراسي
الأسبوع 1	مقدمة عن البرمجيات وهندسة البرمجيات
الأسبوع 2	أهداف هندسة البرمجيات
الأسبوع 3	-تصنيف البرمجيات -بناءً على نوع التطبيق -مجالات تطبيق البرمجيات
الأسبوع 4	عملية البرمجيات -أنواع أنشطة دورة حياة تطوير البرمجيات
الأسبوع 5	نماذج دورة حياة البرمجيات
الأسبوع 6	-تطوير Agile من وجهة نظر الصناعة -فوائد نهجAgile
الأسبوع 7	-مبادئ Agile -خمس ممارسات أساسية
الأسبوع 8	-هندسة المتطلبات -المتطلبات الوظيفية وغير الوظيفية
الأسبوع 9	عمليات هندسة المتطلبات
الأسبوع 10	-استخراج المتطلبات -توثيق المتطلبات
الأسبوع 11	-التحقق من صحة المتطلبات -تعديل المتطلبات
الأسبوع 12	التصميم والتنفيذ
الأسبوع 13	قضايا التنفيذ
الأسبوع 14	-متى نستخدم مخططات الصفوف (Class Diagrams) -تدوين الصفوف في UML -خصائص الصفوف -عمليات الصفوف (Methods) -مدى الوصول للصفوف(Visibility)
الأسبوع 15	-العلاقة (Association) -الوراثة (Inheritance) -التجميع (Aggregation) -التركيب (Composition) -الاعتمادات والقيود (Dependencies & Constraints) -التحقيق (Realization) -الكاردينالية(Cardinality)
الأسبوع 16	أسبوع تحضيري قبل الامتحان النهائي

خطة التدريس (المنهج الأسبوعي)	
	المنهج الدراسي
الأسبوع 1	مقدمة لأدوات هندسة البرمجيات: التعرف على الأدوات المستخدمة طوال المقرر، بما في ذلك Jira وGitHub وأي بيئات تطوير متكاملة (IDEs) ذات صلة
الأسبوع 2	التحكم بالإصدارات باستخدام GitHub: تعلم أساسيات التحكم بالإصدارات، بما في ذلك الالتزام بالتغييرات (Commit)، والفروع (Branching)، والدمج (Merging)
الأسبوع 3	GitHubالمتقدم: استكشاف ميزات متقدمة مثلPull Requests ، مراجعة الشيفرة(Code Reviews) ، وحل التعارضات (Conflict Resolution)
الأسبوع 4	مقدمة إلى Agile وScrum: تعلم المبادئ وكيفية استخدامها في تطوير البرمجيات
الأسبوع 5	العمل مع User Stories في Jira: تعلم كيفية إنشاء وإدارة Stories User وكيفية دمجها في عملية تطوير Agile
الأسبوع 6	تخطيط Sprint باستخدام Jira وScrum: تعلم كيفية تقدير الجهد، ترتيب أولويات المهام، وتوزيع المهام على أعضاء الفريق
الأسبوع 7	إدارة Sprint باستخدام Scrum: محاكاة تشغيل Sprint، بما في ذلك الاجتماعات اليومية (Daily Stand-ups)، متابعة التقدم باستخدام Burndown Chart، وإدارة التغييرات في Sprint
الأسبوع 8	الاختبار والتكامل المستمر باستخدام GitHub: تعلم تنفيذ الاختبارات الآلية والتكامل المستمر باستخدام GitHub Actions
الأسبوع 9	مراجعة Sprint وعقد Retrospective: تعلم كيفية إجراء مراجعة Sprint واستخدام النتائج لتحسين Sprints المستقبلية
الأسبوع 10	Jiraالمتقدم: استكشاف ميزات متقدمة مثل Workflows مخصصة، البحث المتقدم، والتقارير
الأسبوع 11	تخطيط الإصدار باستخدام Agile وScrum: تعلم كيفية تخطيط إصدار برمجي، بما في ذلك تحديد النطاق، جدولة الإصدار، وإدارة المخاطر
الأسبوع 12	إدارة مشروع برمجي باستخدام Agile وJira وGitHub: محاكاة إدارة مشروع من البداية للنهاية
الأسبوع 13	التعاون ومراجعة الشيفرة باستخدام GitHub: تعلم التعاون الفعال مع الفريق، إجراء مراجعات الشيفرة، والتعامل مع الملاحظات
الأسبوع 14	توسيع Scrum باستخدام Jira: تعلم كيفية توسيع Scrum للمشاريع والفرق الكبيرة، بما في ذلك استخدام تقنيات مثل Scrum of Scrums
الأسبوع 15	المشروع النهائي: تطبيق كل ما تم تعلمه في مشروع نهائي، قد يشمل تطوير تطبيق برمجي باستخدام Agile وJira وGitHub

المصادر التعليمية والتدريسية		
	النص	متوفر في المكتبة؟
الكتب الأساسية / المطلوبة	SOFTWARE ENGINEERING, Ninth Edition Ian Sommerville, Addison-Wesley, ISBN 10: 0-13-703515-2 ISBN 13: 978-0-13-703515-1.	نعم
الكتب الموصي بها	P. A. Laplante and M. Kassab, What every engineer - ,should know about software engineering. CRC Press .2022 ,J. Bosch, H. H. Olsson, and I. Crnkovic - ",Engineering ai systems: A research agenda“ ,.Artif. Intell. Paradig. Smart Cyber-Physical Syst pp. 1–19, 2021	لا
المواقع الإلكترونية		

مخطط الدرجات				
المجموعة	الدرجة	التقدير	التقدير%	التقدير
مجموعة النجاح (100 - 50)	A - ممتاز	امتياز	90 - 100	أداء ممتاز
	B-جيد جداً	جيد جدا	80 - 89	فوق المتوسط مع بعض الأخطاء
	C-جيد	جيد	70 - 79	عمل جيد مع أخطاء ملحوظة
	D- مقبول	متوسط	60 - 69	مقبول لكن مع نقائص كبيرة
	E - كافٍ / مرضٍ	مقبول	50 - 59	العمل يلبي الحد الأدنى من المعايير
مجموعة الرسوب (49 - 0)	FX-راسب (قيد المعالجة)	راسب (قيد المعالجة)	(45-49)	يتطلب مزيدًا من العمل ولكن يُمنح الطالب الدرجة
	F-راسب	راسب	(0-44)	يتطلب قدرًا كبيرًا من العمل

ملاحظة:

سيتم تقريب العلامات العشرية التي تزيد أو تقل عن 0.5 إلى العلامة الكاملة الأعلى أو الأدنى (على سبيل المثال، العلامة 54.5 سيتم تقريبها إلى 55، بينما العلامة 54.4 سيتم تقريبها إلى 54). تطبق الجامعة سياسة عدم قبول حالات الرسوب القريبة من النجاح، لذا فإن التعديل الوحيد للدرجات الممنوحة من قبل المصحح/المصححين الأصليين سيكون التقريب التلقائي الموضح أعلاه فقط.