نموذج وصف المقرر الدراسي

معلومات المقرر الدراسية				
اسم المقرر	اساسيات برمجة 2		أسلوب التدريس	
نوع المقرر	رئيسية		☒ محاضرة	
رمز المقرر	**IT203**		☒ عملي	
عدد الوحدات	7			
عدد ساعات المقرر	175			
مستوى المقرر الدراسي	الأول	الفصل الدراسي	2	
القسم الأكاديمي	تكنولوجيا المعلومات	الكلية	كلية العلوم	
مسؤول المادة	ا.م.د مجسن حسن حسين	الايميل	mohsin.h@uokerbala.edu.iq	
اللقب العلمي	أستاذ مساعد	الشهادة الاكاديمية	دكتوراه	
مدرس المادة	ا.م.د مجسن حسن حسين	الايميل	mohsin.h@uokerbala.edu.iq	
اسم مراجع المقرر الدراسي	ا.م.د حيدر محمد علي	الايميل	hayder.alghanami@uowa.edu.iq	
تاريخ موافقة اللجنة العلمية	2024-1-20	اصدار	V1	

العلاقة مع المقررات الدراسية الاخرى			
المتطلب السابق للمادة	اساسيات برمجة 1	الفصل الدراسي	1
المتطلبات المصاحبة للمادة	-	الفصل الدراسي	-

جامعة وارث الانبياء
كلية العلوم
قسم تكنولوجيا المعلومات

جامعة وارث الانبياء
عمادة كلية العلوم

مصادقة رئيس القسم

مصادقة السيد عميد الكلية المحترم

أهداف المادة، ومخرجات التعلم، والمحتوى الإرشادي

هدف المادة الدراسية	فيما يلي الأهداف والفوائد الرئيسية لدراسة مبادئ البرمجة (**Programming Fundamentals II**) بشكل نقاط: 1. حل المشكلات (**Problem Solving**): ○ تعليم الطلبة كيفية تحليل المشكلات وتطوير الخوارزميات لحلها. ○ التركيز على تقنيات حل المشكلات وتصميم الخوارزميات. ○ تعلم تجزئة المشكلات المعقدة إلى أجزاء أصغر وأسهل في الإدارة. 2. تصحيح الأخطاء والاختبار (**Debugging and Testing**): ○ تعليم الطلبة كيفية تصحيح الأخطاء في البرامج واختبارها. ○ استعراض تقنيات اكتشاف الأخطاء وأدوات التصحيح. ○ تعلم استراتيجيات كتابة حالات اختبار فعالة لضمان جودة البرامج. 3. هياكل البيانات (**Data Structures**): ○ التعرف على هياكل البيانات الأساسية مثل: ▪ المصفوفات (**Arrays**) ▪ المكدسات (**Stacks**) ▪ قوائم الانتظار (**Queues**) ▪ الهياكل (**Structures**) ○ دراسة خصائص هذه الهياكل وكيفية تنفيذها واستخدامها في حل مشكلات البرمجة. 4. البرمجة المعيارية (**Modular Programming**): ○ تعلم مفهوم البرمجة المعيارية وتنظيم الأكواد. ○ استخدام الدوال (**Functions**) وتمرير المعاملات (**Parameter Passing**). ○ التركيز على إعادة استخدام الأكواد وكتابة برامج قابلة للصيانة. 5. أفضل ممارسات البرمجة (**Programming Best Practices**):

	○ الالتزام بـ المعايير المهنية في كتابة الأكواد. ○ تعلم توثيق الأكواد بشكل واضح. ○ اتباع اتفاقيات التسمية (**Naming Conventions**). ○ تحسين الأكواد باستخدام تقنيات تحسين الأداء (**Code Optimization**). 6. إدخال وإخراج الملفات (**Files Input and Output**): ○ التعامل مع عمليات الإدخال والإخراج القياسية. ○ القراءة والكتابة من وإلى الملفات. ○ التعرف على أساليب معالجة الأخطاء أثناء عمليات الإدخال والإخراج. 7. مقدمة في البرمجة كائنية التوجه (**Introduction to OOP**): ○ التعرف على المبادئ والمفاهيم الأساسية للبرمجة كائنية التوجه. ○ تعلم إنشاء واستخدام الأصناف (**Classes**) كأساس في البرمجة الحديثة.
مخرجات تعلم المادة الدراسية	فيما يلي بعض المخرجات التعليمية الشائعة لمقرر مبادئ البرمجة (**Programming Fundamentals II**): 1. كتابة الأكواد بفعالية: كتابة أكواد واضحة، ومنظمة، وسهلة القراءة، تتبع معايير كتابة الأكواد وأفضل الممارسات، بما في ذلك التنسيق الصحيح، واستخدام أسماء متغيرات معبّرة، وإضافة التعليقات المناسبة. 2. استخدام هياكل البيانات: تطبيق هياكل البيانات المناسبة مثل المصفوفات، القوائم المرتبطة، المكدسات، وقوائم الانتظار، لتخزين البيانات ومعالجتها بفعالية في المشكلات البرمجية. 3. التصميم المعياري وإعادة الاستخدام: تصميم وتنفيذ برامج معيارية من خلال تقسيمها إلى دوال أو أساليب قابلة لإعادة الاستخدام، مما يسهل إعادة استخدام الأكواد، ويحسن إمكانية صيانتها، ويعزز ممارسات هندسة البرمجيات الجيدة. 4. مهارات تصحيح الأخطاء والاختبار: استخدام تقنيات وأدوات التصحيح لاكتشاف الأخطاء وإصلاحها في البرامج، وتطوير حالات اختبار فعّالة، وإجراء الاختبارات لضمان صحة البرامج وموثوقيتها. 5. فهم مفاهيم البرمجة كائنية التوجه (**OOP**): التعرف على المبادئ والمفاهيم الأساسية للبرمجة كائنية التوجه.
المحتوى الإرشادي	المحتويات الإرشادية لمقرر مبادئ البرمجة (Programming Fundamentals II) تتضمن قائمة بالموضوعات الشائعة كما هو موضح أدناه:

	1. البرمجة المعيارية (Modular Programming): [25 ساعة] ● الدوال والإجراءات (Functions and Procedures). ● نطاق وعمر المتغيرات (Scope and Lifetime of Variables). ● آليات تمرير المعاملات (Parameter Passing Mechanisms). 2. هياكل البيانات (Data Structures): [25 ساعة] ● المصفوفات (Arrays). ● السلاسل النصية والقوائم (Strings and Lists). ● الهياكل (Structures). ● المكدسات وقوائم الانتظار (Stacks and Queues). 3. الإدخال والإخراج (Input and Output): [15 ساعة] ● الإدخال والإخراج القياسي (Standard Input/Output). ● القراءة والكتابة من وإلى الملفات (Reading from and Writing to Files). ● معالجة الأخطاء والاستثناءات (Error Handling and Exception Handling). 4. تصحيح الأخطاء والاختبار (Debugging and Testing): [20 ساعة] ● الأنواع الشائعة للأخطاء البرمجية (Common Types of Programming Errors). ● تقنيات وأدوات التصحيح (Debugging Techniques and Tools). 5. مفاهيم البرمجة كائنية التوجه (Object-Oriented Programming Concepts): [5 ساعات] ● الأصناف والكائنات (Classes and Objects).

استراتيجيات التعليم والتعلم	
استراتيجيات	لتدريس مقرر مبادئ البرمجة (**Programming Fundamentals II**)، يمكن استخدام استراتيجيات متنوعة لتعزيز التعلم الفعّال وزيادة تفاعل الطلبة. فيما يلي بعض استراتيجيات التعلم والتدريس الشائعة: 1. المحاضرات (**Lectures**): ○ تقديم المحاضرات لشرح المفاهيم النظرية والمبادئ والمعرفة الأساسية في مقرر مبادئ البرمجة II. ○ يمكن أن تتضمن المحاضرات وسائل بصرية، وأمثلة عملية، وعروض توضيحية لتعزيز الفهم. 2. المناقشات التفاعلية (**Interactive Discussions**): ○ تشجيع الطلبة على المشاركة الفعالة في المناقشات من خلال طرح الأسئلة، ومشاركة أفكارهم، والتعلم من زملائهم. ○ يمكن أن تركز المناقشات على المفاهيم الصعبة، والتطبيقات الواقعية، أو دراسات الحالة المتعلقة بمبادئ البرمجة II. 3. الجلسات العملية (**Hands-on Lab Sessions**):

	○ تنظيم جلسات عملية تمنح الطلبة فرصة التطبيق العملي للمفاهيم التي تعلموها. ○ تتضمن هذه الجلسات تنفيذ تمارين برمجية واستخدام أوامر البرمجة لتعزيز الفهم وتطوير المهارات العملية. 4. المشاريع الجماعية (**Group Projects**): ○ تكليف الطلبة بمشاريع جماعية تتضمن تصميم، وتنفيذ، وتقييم مكونات برمجية ضمن مقرر مبادئ البرمجة II. ○ تهدف هذه المشاريع إلى تعزيز العمل الجماعي، وحل المشكلات، والتطبيق العملي للمفاهيم المكتسبة. 5. المصادر والدروس الإلكترونية (**Online Resources and Tutorials**): ○ توفير مصادر تعليمية إلكترونية، ودروس تفاعلية، ومواد تعليمية عبر الإنترنت مرتبطة بالمقرر. ○ تتيح هذه الموارد للطلبة الاطلاع على محتوى إضافي، وتعزيز فهمهم، وتقييم تقدمهم بشكل ذاتي. 6. التقييم والتغذية الراجعة (**Assessments and Feedback**): ○ استخدام طرق تقييم متنوعة مثل الاختبارات القصيرة، والواجبات، والمشاريع، والامتحانات. ○ تقديم تغذية راجعة بنّاءة وفي الوقت المناسب لمساعدة الطلبة على تحسين معرفتهم ومهاراتهم.

حمل عمل الطالب			
الساعات المجدولة (ساعات/فصل دراسي)	75	الساعات المجدولة (ساعات/أسبوع)	5
الساعات غير المجدولة (ساعات/فصل دراسي)	97	الساعات غير مجدولة (ساعات/أسبوع)	6.5
الإجمالي (ساعات/فصل دراسي)	172 + 3 فاينل = 175		

تقييم المقرر الدراسي					
		الوقت/العدد	الوزن (الدرجات)	الأسابيع	مخرجات التعلم
التقويم التكويني	اختبارات	5	1% (5)	جميع الاسابيع	1,2,3,4
	واجبات داخل الكلية	5	1%(5)	جميع الاسابيع	جميع المخرجات
	المختبر	5	4% (20)	جميع الاسابيع	جميع المخرجات
	واجبات بيتية	5	2%(10)	جميع الاسابيع	جميع المخرجات
التقييم النهائي	امتحان المد	2 ساعة	10% (10)	9	
	امتحان النهائي	3 ساعة	50% (50)	17	
إجمالي التقييم			%100 (100 درجة)		

خطة التدريس (المنهج النظري الأسبوعي)	
	المنهج الدراسي
الأسبوع 1	الدوال (Functions)
الأسبوع 2	أنواع الدوال (Function Types)
الأسبوع 3	مفهوم الاستدعاء الذاتي (Recursion)
الأسبوع 4	المصفوفة (Array)
الأسبوع 5	المصفوفة أحادية البعد (D Array1)
الأسبوع 6	المصفوفة ثنائية البعد (D Array - Matrix2)
الأسبوع 7	مصفوفة الحروف (Strings)
الأسبوع 8	معالجة النصوص (String Processing)
الأسبوع 9	الامتحان النصفي (**Midterm Exam**)
الأسبوع 10	المصفوفات والدوال (Arrays and Functions)
الأسبوع 11	الهياكل (Structures)
الأسبوع 12	مصفوفة من الهياكل والهياكل المتداخلة (Array of Structures and Nested Structures)
الأسبوع 13	المكدس وقائمة الانتظار (Stack and Queue)
الأسبوع 14	المؤشرات (Pointers)
الأسبوع 15	الملفات (Files)
الأسبوع 16	الأسبوع التحضيري قبل الامتحان النهائي

خطة التدريس (المنهج العملي الأسبوعي)	
	المنهج الدراسي
الأسبوع 1	كتابة الأكواد باستخدام الدوال ((Writing Codes using Functions
الأسبوع 2	كتابة الأكواد باستخدام أنواع الدوال ((Writing Codes using Function Types
الأسبوع 3	كتابة الأكواد باستخدام مفهوم الاستدعاء الذاتي ((Writing Codes using The Concept of Recursion
الأسبوع 4	كتابة الأكواد باستخدام المصفوفات ((Writing Codes using Arrays

الأسبوع 5	كتابة الأكواد باستخدام المصفوفات أحادية البعد ((Writing Codes using 1D Arrays
الأسبوع 6	كتابة الأكواد باستخدام المصفوفات ثنائية البعد (((Writing Codes using 2D Array (Matrix) Matrices)
الأسبوع 7	كتابة الأكواد باستخدام مصفوفة الحروف (((Writing Codes using Array of Characters (Strings) Strings)
الأسبوع 8	كتابة الأكواد باستخدام معالجة النصوص ((Writing Codes using String Processing
الأسبوع 9	الامتحان النصفي ((**Midterm Exam**)
الأسبوع 10	كتابة الأكواد باستخدام المصفوفات والدوال ((Writing Codes using Arrays and Functions
الأسبوع 11	كتابة الأكواد باستخدام الهياكل ((Writing Codes using Structures
الأسبوع 12	كتابة الأكواد باستخدام مصفوفة الهياكل والهياكل المتداخلة (Writing Codes using Array of Structures and Nested Structures)
الأسبوع 13	كتابة الأكواد باستخدام المكدس وقائمة الانتظار ((Writing Codes using Stack and Queue
الأسبوع 14	المؤشرات ((Pointers
الأسبوع 15	الملفات ((Files

المصادر التعليمية والتدريسية		
	النص	متوفر في المكتبة؟
الكتب الأساسية / المطلوبة	**C++: The Complete Reference, Fourth Edition, Herbert Schildt.**	نعم
الكتب الموصي بها	**The C++ Programming Language , Third Edition , Bjarne Stroustrup.**	نعم
المواقع الإلكترونية	**https://stackoverflow.com/**	

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.