نموذج توصيف مقرر: برمجة 1 (Programming 1)

1. معلومات أساسية عن المقرر

البند	التفاصيل
اسم المقرر	برمجة 1
رمز المقرر	SWE121
الساعات المعتمدة	3 نظري +1 عملي
المستوى الدراسي	مستوى أول
المتطلبات السابقة	مدخل إلى الخوارزميات والبرمجة
لغة التدريس	عربي + انكليزي
تاريخ آخر تحديث	9/12/2025

2. أهداف المقرر (Course Objectives)

- الهدف العام:

تهيئة الطالب لفهم المبادئ الأساسية للبرمجة باستخدام لغة C++، وتطوير مهارات التحليل المنطقي، وحل المشكلات، وبناء برامج متكاملة تتضمن هياكل بيانات بسيطة وصولاً إلى البرمجة الكائنية OOP.

- الأهداف التفصيلية (SMART):

بنهاية المقرر سيكون الطالب قادرًا على:

1. كتابة برامج تستخدم المتغيرات وأنواع البيانات والتعبيرات بشكل صحيح.
2. التعامل مع المصفوفات Arrays والعمليات الأساسية عليها.
3. تصميم دوال Functions قابلة لإعادة الاستخدام، وتمرير المعطيات بالقيمة وبالمرجع.
4. استخدام المؤشرات pointers في إدارة الذاكرة البسيطة.
5. بناء برامج تعتمد على البرمجة الكائنية: إنشاء الأصنافClasses ، الخصائص، الدوال الأعضـاء.
6. استخدام بيئة تطوير مناسبة مثل Code::Blocks أو Visual Studio Code لكتابة البرامج وتصحيح الأخطاء.

3. مخرجات التعلم (Learning Outcomes – LOs)

المخرج التعليمي	المعيار العالمي
يتعرف على مكونات البرنامج في C++ وأنواع البيانات الأساسية.	Dublin – Knowledge: فهم المعرفة الأساسية في المجال. ABET 1: القدرة على اكتساب المعرفة الهندسية الأساسية.
يشرح آلية عمل المصفوفات وتمثيلها في الذاكرة.	Dublin – Knowledge: فهم المبادئ النظرية. ABET 1: تطبيق مفاهيم علمية لحل مشكلات تقنية.
يوضح مفهوم المؤشر Pointer وعلاقته بالمصفوفات والذاكرة.	Dublin – Knowledge: تحليل العلاقات بين مكوّنات المعرفة. ABET 2: القدرة على تحليل المشكلات التقنية وتفسير بنيتها.
يفهم المبادئ الأولى للبرمجة كائنية التوجه OOP.	Dublin – Knowledge: استيعاب المفاهيم الحديثة في الحوسبة. ABET 1 & 6: دمج المعرفة الهندسية في الحلول البرمجية.
يحلل المشكلات ويحولها إلى خوارزميات قابلة للتنفيذ.	Dublin – Skills: القدرات المعرفية في التحليل. ABET 2: تحليل المشكلات الهندسية وتطوير حلول مناسبة.

يختار بنية التحكم المناسبة لحل المشكلة (شرطية– تكرارية).	Dublin – Skills: اتخاذ قرارات مبنية على فهم تقني. ABET 2: تحديد واختيار الأدوات والمنهجيات المناسبة.
يكتب وينفذ برامج بلغة C++ تشمل الإدخال/الإخراج.	Dublin – Applied Skills: تطبيق المهارات العملية. ABET 6: تطوير التطبيقات البرمجية باستخدام أدوات مهنية.
ينشئ برامج تتضمن مصفوفات ومؤشرات ودوال وأصناف.	Dublin – Applied Skills: تكامل المفاهيم في التطبيق. ABET 6: القدرة على بناء أنظمة برمجية متكاملة.
يختبر البرامج ويصحح الأخطاء.	Dublin – Skills: التقييم والتحسين. ABET 4: القدرة على تفسير النتائج وإجراء التعديلات.
يعمل بفعالية ضمن مجموعات صغيرة.	Dublin – Competence: العمل الجماعي والمسؤولية. ABET 5: تطوير مهارات العمل الجماعي والاتصال.
يستخدم التفكير المنهجي والمنطقي في البرمجة.	Dublin – Competence: اتخاذ قرارات مهنية مبنية على تحليل. ABET 2 & 4: التفكير النقدي واتخاذ القرار المبني على الأدلة.

4. محتوى المقرر (مُفصَّل حسب الأسابيع)

الأسبوع	الموضوع	الأنشطة	المواد الداعمة
1	مقدمة إلى البرمجة و++C – بناء البرنامج – المتغيرات – أنواع البيانات -العبارات الشرطية- الحلقات التكرارية	أمثلة داخل الصف – حل مسائل بسيطة	cppreference (Basics)
2	المصفوفات الأحادية – Arrays تعريف – إدخال – عمليات أساسية	تطبيقات على المصفوفات	نماذج أكواد
3	مصفوفات ثنائية الأبعاد – الخوارزميات عليها	تطبيقات على المصفوفات	نماذج أكواد
4	النصوص والسلاسل المحرفيةstrings	تطبيقات على السلاسل المحرفية	نماذج أكواد
5	الدوال – Functions القيمة والمرجع	تطبيقات على الدوال	نماذج أكواد
6	الدوال العودية Recursion	تطبيقات على العودية	نماذج أكواد
7	المؤشرات Pointers – التعامل مع الذاكرة – العلاقة مع المصفوفات	تطبيقات على المؤشرات	نماذج أكواد
8	المؤشرات والدوال – المصفوفات والدوال	تطبيقات على المؤشرات والدوال والمصفوفات	نماذج أكواد
9	الهياكل Structs – مقدمة OOP	تطبيقات على الهياكل	نماذج أكواد
10	تعريف الصفوف Classes – المتغيرات الأعضاء – الدوال الأعضاء	تطبيقات على الصفوف	نماذج أكواد
11	الباني Constructor – الهادم Destructor – المبادئ الأساسية للتغليف	تطبيقات على الصفوف وتوابعها	نماذج أكواد

7. طرق التدريس والتعلم

- الطرق المستخدمة:

- محاضرات تفاعلية مدعمة بالأمثلة.
- جلسات مخبرية باستخدام بيئة تطوير.
- مناقشة وتحليل حالات(Problem Solving)
- مشاريع قصيرة فردية/جماعية.

- التكنولوجيا الداعمة:

- محررات(C++ (Code::Blocks / VS Code
- عرض شرائح تعليمية.

6. تقييم التعلم (Assessment Methods)

نوع التقييم	الوصف	النسبة	المخرجات المقاسة
امتحان نصفي	مسائل برمجية وأسئلة تحليلية	%20	فهم الأساسيات والمصفوفات
تحليل حالة	يُعطى الطلاب سيناريو/مشكلة واقعية تتطلب تحليل المتطلبات وتصميم خوارزمية واختيار البنى المناسبة (Arrays, Functions, Classes) ثم تقديم حل منطقي مكتوب دون تنفيذ. يقيّم قدرة الطالب على التفكير المنطقي وتصميم الحل قبل البرمجة.	%10	LOs: مهارات التحليل، تصميم الخوارزميات، اختيار بنى البيانات
مشروع جماعي	برنامج متكامل يتضمن Arrays + Functions + Class	%10	مهارات عملية
اختبارات قصيرة	تطبيقات قصيرة داخل الصف والمخبر	%5	مؤشرات، دوال
المشاركة الصفية	حضور، تفاعل، نشاط	%5	LO-Soft Skills
امتحان نهائي	أسئلة تحليل، تتبع برامج، كتابة كود	%50	جميع المخرجات

7. المراجع والموارد

المراجع الأساسية

1. Bjarne Stroustrup – *Programming: Principles and Practice Using C++*
   https://www.stroustrup.com/programming.html
2. Bjarne Stroustrup – *The C++ Programming Language*
   https://www.stroustrup.com

الموارد الإلكترونية

- cppreference (مرجع رسمي للغة)
  https://en.cppreference.com
- TutorialsPoint C++ Guide
  https://www.tutorialspoint.com/cplusplus

أدوات عملية

- Code::Blocks
- Visual Studio Code

استاذ المقرر: م. مايا تقي

التوقيع