MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information معلومات المادة الدراسية					
Module Title	**Basics of Object-Oriented Programming**			**Module Delivery**	
Module Type	**C**			☒ **Theory** ☒ **Lecture** ☒ **Lab** ☒ **Tutorial** ☒ **Practical** ☒ **Seminar**	
Module Code	**AI2103**				
ECTS Credits	**7**				
SWL (hr/sem)	**175**				
Module Level	2		**Semester of Delivery**	1	
Administering Department	AI		**College**	Computer Science and Information Technology	
Module Leader	Abdulkareem Zwaen		**e-mail**	abdulkareem@uowa.edu.iq	
Module Leader's Acad. Title	lecturer		**Module Leader's Qualification**	Ph.D	
Module Tutor	Abdulkareem Zwaen		**e-mail**	abdulkareem@uowa.edu.iq	
Peer Reviewer Name	Hayder Mohammad		**e-mail**	Hayder.alghanimi@uowa.edu.iq	
Scientific Committee Approval Date	01/09/2026		**Version Number**	1.0	

Relation with other Modules العلاقة مع المواد الدراسية الأخرى			
Prerequisite module	---	**Semester**	--
Co-requisites module		**Semester**	

Department Head Approval **Dean of the College Approval**

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Objectives أهداف المادة الدراسية	1. Introduce the principles of object-oriented programming in a higher-level programming language in c++. 2. Analyze a problem statement to develop a mental model of objects necessary to create a software architecture 3. Utilize object-oriented programming to frame software architectures, with care towards separation of concerns and abstraction 4. Gain skills in designing, and programming software for reuse of code. 5. Establish development methods in object-oriented programming to qualify students for teaching the language in other settings
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	1. Explain the motivation for and development of object-oriented programming languages. 2. Produce a set of use cases given a problem statement. 3. Produce class diagrams, object interaction diagrams and object state transition diagrams for a given problem. 4. Describe the essential features of an object-oriented programming language. 5. Produce and/or debug code fragments that illustrate principles of object-oriented software development. 6. Describe the principles for testing object-oriented software and derive sets of test data given a specification.
Indicative Contents المحتويات الإرشادية	✓ Structured programming ✓ Procedural programming ✓ Abstract data types (ADTs) ✓ Type of variable and range of applicable operations ✓ Use of classes in object-oriented programming ✓ How coupling and cohesion are implemented in OOP ✓ Abstraction and Encapsulation ✓ Data hiding/information hiding ✓ Classes and objects instances

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	There are different teaching and learning activities including lectures and laboratories. The concepts, process, and applications of data science will be discussed in lectures. Students will also learn computer programming knowledge and the skills of manipulating, processing, retrieving, storing, and plotting data. Students will develop small programs and learn different in laboratories.

Student Workload (SWL) الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا			
Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	108	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	7
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	67	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	5
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	**175**		

Module Evaluation تقييم المادة الدراسية					
		Time/Number	**Weight (Marks)**	**Week Due**	**Relevant Learning Outcome**
Formative assessment	**Quizzes**	5	10% (10)	2,4,6,9,12	LO #1, #2 and #6
	Assignments	5	10% (10)	3,5,8,11,13	LO #3, #4 and #6
	Projects / Lab.	1	10% (10)	Continuous	All
	Report	1	10% (10)	13	LO #6
Summative assessment	**Midterm Exam**	2hr	10% (10)	7	LO #1 - #6
	Final Exam	3hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus) المنهاج الاسبوعي النظري	
	Material Covered
Week 1	Overview of C++ Programming
Week 2	Introduction to Object Oriented Programming
Week 3	Comparison between Structural programming and OOP.
Week 4	Class in OOP
Week 5	Class Object
Week 6	Class constructor
Week 7	Local variable and class variable

Week 8	Static and none static variable.
Week 9	Static and none static method.
Week 10	Encapsulation
Week 11	Inheritance (Super class and sub class)
Week 12	Mid exam
Week 13	Array Objects & Type of inheritance
Week 14	OOP Project
Week 15	**Preparatory week before the final Exam**

Delivery Plan (Weekly Lab. Syllabus) المنهاج الاسبوعي للمختبر	
	Material Covered
Week 1	Lab 1: develop a program to implement 1 dimension array
Week 2	Lab 2: develop a program to perform matrix operation using multi-dimensions array
Week 3	Lab 3: develop program that implement a class and use it with objects
Week 4	Lab 4: develop program that implement a class and create array of objects
Week 5	Lab 5: write program for single inherence
Week 6	Lab 6: write program for hybrid inherence
Week 7	Lab 7: write code for overload function
Week 8	**Lab 8:** Write code for Operator Overloading (Overloading unary and binary operators)
Week 9	**Lab 9:** Implement Static Data Members, Static Methods, and Data Encapsulation
Week 10	**Lab 10:** Implement Polymorphism, Function Overriding, and Virtual Functions
Week 11	**Lab 11:** Implement Abstraction using Pure Virtual Functions and Abstract Classes
Week 12	**Lab 12:** Mid-term Practical Exam
Week 13	**Lab 13:** Implement Pointers to Objects and Dynamic Memory Allocation (new/delete)
Week 14	**Lab 14:** Hands-on development and implementation of the OOP Capstone Project
Week 15	Lab 15: Final Practical Evaluation, Code Review, and Project Submissions

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	**Available in the Library?**
Required Texts	Object-oriented programming with C++ by E.Balagurusamy, 2nd Edition, TMH.	No
Recommended Texts	1. Object Oriented Design by Rumbaugh (Pearson publication) 2. Object-oriented programming in Turbo C++ By Robert Lafore, Galgotia Publication.	No
Websites	https://www.coursera.org	

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.