MODULE DESCRIPTION FORM
نموذج وصف المادة الدراسية

Module Information	معلومات المادة الدراسية		
Module Title	**Structured Programming**	**Module Delivery**	
Module Type	**C**	☒ Theory	
Module Code	**AI1203**	☒ Lecture	
ECTS Credits	**8**	☒ Lab	
SWL (hr/sem)	**200**	☒ Tutorial ☒ Practical ☒ Seminar	
Module Level	1	**Semester of Delivery**	2
Administering Department	AI	**College** CSIT	
Module Leader	Ali Kaream	**e-mail** ali.kaream@uowa.edu.iq	
Module Leader's Acad. Title	Lecturer	**Module Leader's Qualification**	Ph.D.
Module Tutor	Ali Kaream	**e-mail** ali.kaream@uowa.edu.iq	
Peer Reviewer Name	Ali Mahmoud Ali	**e-mail** ali.mahmoud@uowa.edu.iq	
Scientific Committee Approval Date	01/03/2026	**Version Number**	1.0

Relation with other Modules	العلاقة مع المواد الدراسية الأخرى		
Prerequisite module	None	**Semester**	-
Co-requisites module	None	**Semester**	-
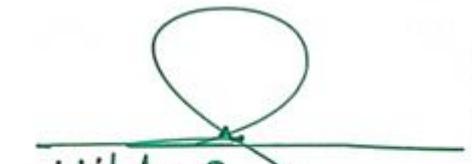
Department Head Approval   **Dean of the College Approval**

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Objectives أهداف المادة الدراسية	Learn how to use the Advanced Tools helps programmers write fast, portable programs The main principles of programming and the development of programming languages Learn the principles of Structure programming
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	Upon successful completion of this module, learners will be able to: 1. **Algorithmic Logic:** Deconstruct complex problems into step-by-step procedural logic using pseudocode and flowcharts before writing a single line of code. 2. **Syntax Proficiency:** Demonstrate mastery of core programming constructs, including variables, data types (integers, floats, strings, Booleans), and basic operators. 3. **Control Structures:** Implement decision-making logic using nested if-else statements and switch cases to handle multiple execution paths. 4. **Iteration & Loops:** Execute repetitive tasks efficiently using for, while, and do-while loops, ensuring proper termination conditions to avoid infinite cycles. 5. **Modular Design:** Break down large programs into reusable, independent functions/procedures to improve readability and reduce code redundancy. 6. **Scope & Memory:** Differentiate between local and global variables and understand how data is passed between functions (Pass-by-Value vs. Pass-by-Reference). 7. **Data Structures (Arrays):** Store and manipulate collections of data using single and multi-dimensional arrays, including basic sorting and searching algorithms. 8. **File I/O Operations:** Develop programs that can read from and write to external text and binary files for persistent data storage. 9. **Debugging & Error Handling:** Utilize systematic debugging techniques and trace tables to identify and resolve logical, syntax, and runtime errors. 10. **Coding Standards:** Apply industry-standard naming conventions, indentation, and documentation (commenting) to produce "clean code" that is easily readable by others. 11. **Project Lifecycle:** Design, code, and test a final integrated application that solves a multi-step business or mathematical problem.
Indicative Contents المحتويات الإرشادية	Introductions to C++ Programming; Introductions to essential computer graphics concepts and theories; Object Oriented programming for 2D graphics; Algorithms design for 2D graphics; Graphic interface creations and implementations.

Learning and Teaching Strategies
استراتيجيات التعلم والتعليم

Strategies	- Daily and weekly quizzes. - Class room activities. - Guiding the student to some electronic websites.

Student Workload (SWL)
الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا

Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	123	Structured SWL (h/w) الحمل الدراسي المنتظم للطالب أسبوعيا	8
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	77	Unstructured SWL (h/w) الحمل الدراسي غير المنتظم للطالب أسبوعيا	5
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل		**200**	

Module Evaluation
تقييم المادة الدراسية

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	**Quizzes**	5	10% (10)	2,4,6,8,11	LO #1, #2 and #10, #11
	Assignments	5	10% (10)	3,5,9,12,13	LO #3, #4 and #6, #7
	Projects / Lab.	1	10% (10)	Continuous	All
	Report	1	10% (10)	13	LO #5, #8 and #10
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

	Material Covered
Week 1	Function
Week 2	Passing Parameters. Passing by Value. Passing by Reference.
Week 3	Pointers
Week 4	Arrays. Array of One Dimension: Declaration of Arrays.

Week 5	Initializing Array Elements
Week 6	Accessing Array Elements
Week 7	Mid-term Exam
Week 8	Read / Write / Process Array Elements.
Week 9	Array of Two Dimension: Declaration of 2D-Arrays
Week 10	Read / Write / Process Array Elements.
Week 11	Member Function of String stdlib Library.
Week 12	Structures. The Three Ways for Declare the Structure.
Week 13	Array of Structures.
Week 14	The Files
Week 15	
Week 16	**Preparatory week before the final Exam**

Delivery Plan (Weekly Lab. Syllabus) المنهاج الاسبوعي للمختبر	
	Material Covered
Week 1	Function
Week 2	Passing Parameters. Passing by Value. Passing by Reference.
Week 3	Pointers
Week 4	Arrays. Array of One Dimension: Declaration of Arrays.
Week 5	Initializing Array Elements
Week 6	Accessing Array Elements
Week 7	Array of Two Dimension: Declaration of 2D-Arrays
Week 8	Read / Write / Process Array Elements.
Week 9	Array of Structures.
Week 10	The Files
Week 11	Member Function of String stdlib Library.
Week 12	Structures. The Three Ways for Declare the Structure.
Week 13	Pointers of Array.
Week 14	The Files
Week 15	**Preparatory week before the final Exam**

<table>
<tr><th colspan="3">Learning and Teaching Resources
مصادر التعلم والتدريس</th></tr>
<tr><td></td><td>Text</td><td>Available in the Library?</td></tr>
<tr><td>Required Texts</td><td>Mastering C++,   shomme's series</td><td>yes</td></tr>
<tr><td>Recommended Texts</td><td>Stroustrup, B. (2013). The C++ Programming Language (4th ed.).
Deitel, P., & Deitel, H. (2020). C++ How to Program (10th ed.).</td><td></td></tr>
<tr><td>Websites</td><td colspan="2">https://www.learncpp.com https://www.w3schools.com/CPP/default.asp</td></tr>
</table>

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.