MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information معلومات المادة الدراسية					
Module Title	**Computer Fundamentals and Programming I**			**Module Delivery**	
Module Type	**Basic learning activities**			☒ **Theory** ☐ **Lecture** ☒ **Lab** ☐ **Tutorial** ☐ **Practical** ☐ **Seminar**	
Module Code	**URCOM111**				
ECTS Credits	**3**				
SWL (hr/sem)	**75**				
Module Level	1		**Semester of Delivery**	1	
Administering Department	Type Dept. Code		**College**	Type College Code	
Module Leader	Buthaina murad kareem		**e-mail**	Buthaina.Murad@alayen.edu.iq	
Module Leader's Acad. Title	Lecturer		**Module Leader's Qualification**	M.SC	
Module Tutor	-		**e-mail**		
Peer Reviewer Name	-		**e-mail**		
Scientific Committee Approval Date	4/12/2024		**Version Number**		

Relation with other Modules العلاقة مع المواد الدراسية الأخرى			
Prerequisite module	None	**Semester**	
Co-requisites module	None	**Semester**	

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Aims أهداف المادة الدراسية	1. **Introduction to Computer Systems:** Understanding the basic structure and functioning of computer systems, including hardware, software, and operating systems. 2. **Fundamentals of Programming:** Learning the foundational concepts of programming such as variables, data types, control structures (if-else, loops), and functions. 3. **C++ Programming Language:** Gaining proficiency in C++ syntax and semantics, including how to write, compile, and debug C++ programs. 4. **Object-Oriented Programming (OOP) Concepts:** Introducing key OOP principles such as classes, objects, inheritance, polymorphism, and encapsulation using C++. 5. **Problem-Solving Skills:** Developing the ability to analyze problems and design effective solutions using algorithms and C++ programming. 6. **Data Structures:** Exploring basic data structures like arrays, pointers, strings, and linked lists within the context of C++. 7. **Software Development:** Understanding the software development process, including writing clear, maintainable code, and using version control systems. 8. **Application of C++:** Applying C++ programming to solve real-world problems, potentially covering topics such as file handling, dynamic memory management, and simple algorithms. 9. **Ethical and Professional Issues:** Discussing ethical considerations in computing and the role of software developers in society.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	**Knowledge and Understanding** 1. **Fundamental Concepts of Computing:** Demonstrate an understanding of the basic components and operation of a computer system, including hardware, software, and operating systems. 2. **Programming Principles:** Explain the core principles of programming, including algorithm development, control structures, data types, and function usage. 3. **C++ Language Proficiency:** Display a thorough understanding of the C++ programming language, including syntax, semantics, and standard libraries. 4. **Object-Oriented Programming:** Understand and apply object-oriented programming concepts, including classes, objects, inheritance, polymorphism, and encapsulation. **Intellectual Skills** 1. **Problem-Solving:** Analyze complex problems and design effective solutions using appropriate algorithms and data structures, implementing them in C++. 2. **Logical Reasoning:** Apply logical reasoning to develop and debug programs,

	identifying and fixing errors systematically. 3. **Algorithmic Thinking:** Develop and implement algorithms for a variety of computational problems, optimizing for efficiency and clarity. **Practical Skills** 1. **C++ Programming:** Write, compile, and execute C++ programs to solve specific problems, demonstrating proficiency in the use of variables, control structures, functions, and classes. 2. **Data Handling and File Management:** Implement file handling operations in C++ and manage data using appropriate data structures, such as arrays, pointers, and linked lists. 3. **Memory Management:** Apply dynamic memory management techniques in C++, including the use of pointers, dynamic allocation, and deal location. 4. **Use of Development Tools:** Utilize modern integrated development environments (IDEs) and debugging tools to write, test, and debug C++ programs effectively. **Transferable Skills** 1. **Communication:** Communicate technical information effectively, both orally and in writing, particularly in the documentation and explanation of programming solutions. 2. **Teamwork:** Collaborate with others in a team environment to solve problems and develop software projects, demonstrating the ability to work effectively as part of a group. 3. **Self-Learning:** Demonstrate the ability to independently learn and apply new programming concepts and tools, adapting to new challenges in the field of computing.
Indicative Contents المحتويات الإرشادية	**Indicative content includes the following :** **1. Introduction to Computer Systems** • Overview of computer hardware components (CPU, memory, storage devices, input/output devices). • Introduction to software (system software, application software). • Basic concepts of operating systems and their functions. • Data representation (binary, hexadecimal, ASCII). **2. Introduction to Programming** • Understanding programming languages and their classifications (low-level vs high-level languages). • Introduction to algorithms and flowcharts. • The software development lifecycle (SDLC). • Overview of Integrated Development Environments (IDEs) for C++. **3. C++ Programming Basics**

	• Introduction to C++ programming language and its history. • Structure of a C++ program. • Syntax and semantics of C++. • Writing, compiling, and executing a simple C++ program. **4. Variables, Data Types, and Operators** • Declaration and initialization of variables. • Basic data types in C++ (int, float, double, char, bool). • Constants and literals. • Arithmetic, relational, logical, and bitwise operators. • Type conversion and casting. **5. Control Structures** • Conditional statements (if, if-else, switch-case). • Looping structures (for, while, do-while). • Nested control structures. • Break, continue, and goto statements. **6. Functions in C++** • Defining and calling functions. • Function prototypes and parameter passing (by value, by reference). • Recursion in C++. • Scope and lifetime of variables. • Inline functions and default arguments.

Learning and Teaching Strategies استراتيجيات التعلم و التعليم	
Strategies	The Learning and Teaching Strategies describe the methods and approaches used to deliver content and ensure students achieve the learning outcomes. Common strategies include: 1. **Lectures:** Provide core knowledge about computers and C++ programming. 2. **Lab Sessions:** Hands-on practice with programming exercises to apply what's learned in lectures. 3. **Assignments:** Individual tasks to write C++ programs, helping build problem-solving skills. 4. 0nsite Assignments: Collaborative Assignments where students work together to create a software application. 5. **Tutorials:** Small group sessions to help with difficult topics and answer questions. 6. **Online Resources:** Access to lecture notes, tutorials, and quizzes for extra practice outside class. 7. **Assessments:** Regular quizzes, assignments, and a final project or exam to measure understanding.

Student Workload (SWL) الحمل الدراسي للطالب			
Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	48	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	3
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	27	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	1.8
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	75		

Module Evaluation تقييم المادة الدراسية					
		Time/Number	**Weight (Marks)**	**Week Due**	**Relevant Learning Outcome**
Formative assessment	**Quizzes**	1	5% (5)	5, 10	LO #2,3, 8,9
	0nsite Assign.	1	5% (5)	Continuous	LO # 1-15
	Online –assign.	2	10% (5)	Continuous	LO # 1-15
	Projects**/ Lab**	1	15% (15)	13	LO # 1-15
	Report	5	5% (1)	4, 8	LO # 4, 8
Summative assessment	**Midterm Exam**	2	10% (10)	6,12	LO # 1-11
	Final Exam	3hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus) المنهاج الاسبوعي النظري	
	Material Covered
Week 1	Introduction - Overview of the course
Week 2	Basics of computer systems: hardware and software, Microsoft Office overview
Week 3	Types of Computer Network
Week 4	Introduction to C++, Writing, compiling, and running first C++ program
Week 5	Algorithms and Flowcharts
Week 6	Input and Output Statements
Week 7	Mid-term Exam 1
Week 8	Constants and Variables
Week 9	Operators in C++
Week 10	Order of Evaluation
Week 11	Selection Statements
Week 12	Switch Selection Statement
Week 13	Loop Statements (Part-1)
Week 14	Loop Statements (Part-2)
Week 15	Mid-term Exam 2
Week 16	Preparatory week before the final Exam

Delivery Plan (Weekly Lab. Syllabus) المنهاج الاسبوعي للمختبر	
	Material Covered
Week 1	Basics of computer systems: hardware and software, Microsoft Office overview
Week 2	Types of Computer Network
Week 3	Introduction to C++, Writing, compiling, and running first C++ program
Week 4	Input Statements
Week 5	Output Statements
Week 6	Constants
Week 7	Variables
Week 8	Operators in C++
Week 9	Order of Evaluation (Part-1)
Week 10	Order of Evaluation (Part-2)
Week 11	Selection Statements
Week 12	Switch Selection Statement
Week 13	Loop Statements (Part-1)
Week 14	Loop Statements (Part-2)
Week 15	Exam

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	**Available in the Library?**
Required Texts	1- "C++ Primer" by Stanley B. Lippman, Josée Lajoie, and Barbara E. Moo: Considered a comprehensive guide for beginners and intermediate learners. 2- "Effective C++" by Scott Meyers: Focuses on best practices and advanced C++ programming techniques.	Yes
Recommended Texts	"C++ Programming Language" by Bjarne Stroustrup: Authored by the creator of C++, it provides an in-depth understanding of the language.	No
Websites	https://www.cplusplus.com: Offers a comprehensive C++ reference and tutorials for beginners. • GeeksforGeeks: Provides a vast collection of C++ programming tutorials, examples, and practice problems. • Stack Overflow: A popular question-and-answer website where you can find solutions to specific C++ programming problems.	

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.