MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information معلومات المادة الدراسية		
Module Title	**Computer Programming and Applications (C++ programming)**	**Module Delivery**
Module Type	**Basic**	☒Theory ☐Lecture ☒Lab ☐Tutorial ☐Practical ☐Seminar
Module Code	**UOMU024055**	
ECTS Credits	**4**	
SWL (hr/sem)	**100**	

Module Level	UGIII	**Semester of Delivery**	5
Administering Department	MIET	**College**	EETC
Module Leader		**e-mail**	zena@mtu.edu.iq
Module Leader's Acad. Title	Lecturer	**Module Leader's Qualification**	MSC
Module Tutor	Awss Jabbar Majeed	**e-mail**	awss_alogaidi@mtu.edu.iq
Peer Reviewer Name	Dr. Maiser Monther adnan	**e-mail**	aws_basil@mtu.edu.iq
Scientific Committee Approval Date	8/11/2023	**Version Number**	2.0

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 Objectives أهداف المادة الدراسية	1. Understanding the fundamental concepts of C++ programming language environment. 2. The students will understand and learn how to use C++ as an effective programming language. 3. The students will be able to solve different mathematical and engineering problems as well as design projects using code. 4. Students will acquire the knowledge of basic C++ syntax such as: variables, input, output, vectors, matrices, functions. 5. The students will gain the necessary skills to design and implement appropriate algorithms that solve problems dealing with different mathematical and engineering applications.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	1. Introducing the history and philosophy of C and C++, how C++ adds generic programming concepts to the C language, programming language standards, the mechanics of creating a program. 2. The students learn how to create a C++ program and the general format for a C++ program 3. The “# include directive”, the main() function, How to use the cout object for output, how to place comments in a C++ Program, how and when to use endl, how to declare and use variables, how to use the “cin” object for input, how to define and use simple functions. 4. Learn rules for naming C++ variables, C++ built-in integer types, numeric constants of various integer types, using the const qualifier to create symbolic constants, C++’s built-in floating-point types, C++’s arithmetic operators, automatic type conversions, forced type conversions (type casts). 5. How to create and use arrays, how to create and use C-style strings. 6. How to create and use string class strings, how to use the “getline()”. 7. Learn how to create and use structures, how to create and use pointers, how to create dynamic arrays, how to create dynamic structures, automatic, static, and dynamic storage. 8. Understand the for loop, expressions and statements, the increment and decrement operators: ++ and --, combination assignment operators, compound statements (blocks). 9. The if statement, the if else statement, logical operators: “&&” ,” ||” , and “!”, the “cctype” library of character functions, the conditional operator, the switch statement, the continue and break statements, number-reading loops, basic file input/output. 10. The C++ view of input and output, the “iostream” family of classes redirection,

	ostream class methods, formatting output, “istream” class methods, stream states, file I/O, using the “ifstream” class for input from files, using the “ofstream” class for output to files, using the “fstream” class file input and output, command-line processing, binary files.
Indicative Contents المحتويات الإرشادية	Indicative content includes the following. Introducing to the history and philosophy of C and of C++, How C++ adds object-oriented, concepts to the C language, How C++ adds generic programming concepts to the C language, programming language standards, and the mechanics of creating a program. How to create a C++ program, the general format for a C++ program the #include directive, the “main()” function, how to use the cout object for output, how to place comments in a C++ program, how and when to use “endl”, how to declare and use variables, how to use the “cin” object for input, and how to define and use simple functions. Rules for naming C++ variables, C++’s built-in integer types, Numeric constants of various integer types, Using the “const” qualifier to create symbolic constants, C++’s built-in floating-point types, C++’s arithmetic operators, automatic type conversions, and forced type conversions (type casts). [15 hrs] How to create and use arrays, how to create and use C-style strings ,how to create and use string class strings, How to use the “getline()” and how to create and use arrays, how to create and use C-style strings, how to create and use string class strings, How to use the “getline(): and “get()” methods for reading strings, How to mix string and numeric input, how to create and use structures, how to create and use pointers, how to create dynamic arrays, and how to create dynamic structures, automatic, static, and dynamic storage. The for loop, Expressions and statements, The increment and decrement operators: ++ and --, Combination assignment operators, Compound statements (blocks), The comma operator, Relational operators: > , >= , == , <= ,< , and !=, The while loop, The do while loop, The get() character input method, The end-of-file condition, Nested loops and two-dimensional arrays.[15 hrs] The if statement, The if else statement, Logical operators: && , || , and !, The cctype library of character functions, The conditional operator: ?:. The switch statement, the continue and break statements, number-reading loops, and basic File input/output [10 hrs] The C++ view of input and output, the “iostream” family of classes Redirection, “ostream” class methods, formatting output, “istream” class methods, stream states, file I/O, using the “ifstream” class for input from files, using the “ofstream” class for output to files, using the “fstream” class file input and output, command-line processing, binary files, and random file access. [15 hrs] Revision problem classes [5 hrs]

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	The main strategy that will be adopted in delivering this module is to encourage students’ participation in the exercises, while at the same time refining and expanding their critical thinking skills. This will be achieved through classes, interactive tutorials and by considering types of simple experiments involving some sampling activities that are interesting to the students.

Student Workload (SWL) الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا			
Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	64	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	4
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	36	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	3
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	**100**		

Module Evaluation تقييم المادة الدراسية					
		Time/Number	**Weight (Marks)**	**Week Due**	**Relevant Learning Outcome**
Formative assessment	**Quizzes**	2	20% (20)	5 and 10	LO #1, #2 and #10,
	Assignments	2	10% (10)	2 and 12	LO #3, #4 and #6, #7
	Project/Lab	5	10% (10)	2 and 13	LO #2, #4 and #5, #9
	Report	N/A			
Summative assessment	**Midterm Exam**	2hr	10% (10)	7	LO #1 - #7
	Final Exam	4hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)

المنهاج الاسبوعي النظري

	Material Covered
Week 1	Introducing history and philosophy of C and C++, How C++ adds object-oriented, concepts to the C language, How C++ adds generic programming concepts to the C language, Programming language standards, and the mechanics of creating a program.
Week 2	How to create a C++ program, general format for a C++ program, "#include" directive, The main() function, how to use the "cout" object for output, how to place comments in a C++ program, how and when to use "endl", how to declare and use variables, how to use the "cin" object for input, and how to define and use simple functions.
Week 3 &4	Rules for naming C++ variables, C++'s built-in integer types, numeric constants of various integer types, using the "const" qualifier to create symbolic constants, C++'s built-in floating-point types, C++'s arithmetic operators, automatic type conversions, and forced type conversions (type casts).
Week 5	How to create and use arrays, how to create and use C-style strings, how to create and use string class strings, how to use the "getline()" and "get()" methods for reading strings,
Week 6	How to mix string and numeric input, how to create and use structures, how to create and use pointers, and how to create dynamic arrays. How to create dynamic structures, automatic, static, and dynamic storage.
Week 7	**Mid-term Exam**
Week 8	The for loop, Expressions and statements, increment and decrement operators: ++ and --, combination assignment operators, compound statements (blocks), comma operator, and relational operators: > , >= , == , <= ,< , and !=.
Week 9	While loop, do "while" loop, and "get()" character input method.
Week 10	The end-of-file condition, nested loops and two-dimensional arrays.
Week 11	if statement, if else statement, logical operators: && , || , and !", "cctype" library of character functions, and conditional operator: ?.
Week 12	"Switch" statement, "continue" and "break" statements, number-reading loops, and basic File input/output.
Week 13 &14	C++ view of input and output, the "io-stream" family of classes redirection, "io-stream" class methods, formatting output, "io-stream" class methods, stream states, file I/O, using the "ifstream" class for input from files, and using the "ofstream" class for output to files.
Week 15	Command-line processing, binary files, random file access, and "Incore" formatting **Preparatory week before the final Exam**

Delivery Plan (Weekly Lab. Syllabus)

المنهاج الاسبوعي للمختبر

	Material Covered
Week 1	Introduction, C++ Environment.
Week 2	The general format for a C++ program, the #include directive, The main() function, how to use the cout object for output, How to place comments in a C++ Program, how and when to use endl, how to declare and use variables, how to use the cin object for input, and how to define and use simple functions.
Week 3	
Week 4	Variables and assignment statement, logical operator.
Week 5	
Week 6	Using the const qualifier to create symbolic constants, C++'s built-in floating-point types, and C++'s arithmetic operators.
Week 7	
Week 8	Arrays, Built in functions, Basic Matrix Functions
Week 9	
Week 10	Control Statements(Conditional statements: If, Else, Elseif, switch case)
Week 11	
Week 12	How to create dynamic structures, automatic, static, and dynamic storage.
Week 13	

Learning and Teaching Resources

مصادر التعلم والتدريس

	Text	Available in the Library?
Required Texts	P.B. Mahapatra "C++"	yes
Recommended Texts	A Complete Guide to Programming in C++ Ulla Kirch-Prinz Peter Prinz	
Websites	https://www.coursera.org/browse/physical-science-and-engineering/electrical-engineeringhttp://www.lmpt.univ-tours.fr/~volkov/C++.pdf	

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.