

وزارة التعليم العالي والبحث العلمي  
جهاز الإشراف والتقويم العلمي  
دائرة ضمان الجودة والاعتماد الأكاديمي

## استمارة وصف البرنامج الأكاديمي للكليات والمعاهد

للعام الدراسي ٢٠٢٣-٢٠٢٤

الجامعة : تكريت

الكلية/ المعهد: كلية هندسة العمليات النفطية

القسم العلمي : هندسة سيطرة المنظومات النفطية

تاريخ ملء الملف : ٢٠٢٣/١١/٢٥

التوقيع :

اسم المعاون العلمي : أ.م.د. عمر ياسين ضايح

التاريخ : ٢٠٢٣/١٢/٣

التوقيع :

اسم رئيس القسم : م. ياسين خضر ياسين

التاريخ : ٢٠٢٣/١١/٢٨

دقق الملف من قبل

شعبة ضمان الجودة والأداء الجامعي

اسم مدير شعبة ضمان الجودة والأداء الجامعي: م.م. أيوب إبراهيم محمد

التاريخ : ٢٠٢٣/١١/٢٨

التوقيع :

مصادقة السيد العميد

أ.م.د. غسان حمد عبد الله

٢٠٢٣/١٢/٣

التوقيع :

# MODULE DESCRIPTION FORM

## نموذج وصف المادة الدراسية

Module Information			
معلومات المادة الدراسية			
Module Title	<b>Computer Engineering Principles</b>		Module Delivery
Module Type	Support		<input checked="" type="checkbox"/> Theory <input type="checkbox"/> Lecture <input checked="" type="checkbox"/> Lab <input type="checkbox"/> Tutorial <input type="checkbox"/> Practical <input type="checkbox"/> Seminar
Module Code	<b>PCS125</b>		
ECTS Credits	6		
SWL (hr/sem)	<b>150</b>		
Module Level	1	Semester of Delivery	
Administering Department	PCS	College	PCSE
Module Leader	Mohammed Rashid Subhi	e-mail	Abo1986hhh@tu.edu.iq
Module Leader's Acad. Title	Assistant Lecturer	Module Leader's Qualification	MS.C.
Module Tutor		e-mail	
Peer Reviewer Name	Name	e-mail	E-mail
Scientific Committee Approval Date	01/06/2023	Version Number	1.0

Relation with other Modules			
العلاقة مع المواد الدراسية الأخرى			
Prerequisite module	PCS112	Semester	1
Co-requisites module	None	Semester	

## Module Aims, Learning Outcomes and Indicative Contents

### أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية

<p><b>Module Objectives</b> أهداف المادة الدراسية</p>	<p>Objective 1: Introduction to C++: Introduce students to the C++ programming language, its syntax, and basic features.</p> <p>Objective 2: Variables and Data Types: Teach students about variables, their types, and how to declare and use them in C++. Cover fundamental data types such as integers, floating-point numbers, characters, and strings.</p> <p>Objective 3: Input and Output: Familiarize students with input and output operations in C++, including reading user input and displaying output on the console.</p> <p>Objective 4: Control Structures: Teach students about control structures in C++, such as if-else statements, switch statements, and loops (while, do-while, for). Explain how these structures can be used to control program flow and make decisions.</p> <p>Objective 5: Functions: Introduce students to functions in C++, including function definition, parameters, return types, and function calls. Teach them how to create and use functions to modularize their code and perform specific tasks.</p> <p>Objective 6: Arrays and structure: Cover the concept of arrays in C++, including declaration, initialization, and accessing array elements. Teach students how to work with one-dimensional and multi-dimensional arrays.</p>
<p><b>Module Learning Outcomes</b> مخرجات التعلم للمادة الدراسية</p>	<p>Learning Outcome for as below:</p> <ol style="list-style-type: none"><li>1. Understand the basic syntax and features of the C++ programming language.</li><li>2. Identify and utilize different variable types in C++, such as integers, floating-point numbers, characters, and strings.</li><li>3. Demonstrate the ability to declare, assign values to, and manipulate variables in C++.</li><li>4. Apply input and output operations to interact with users and display information.</li><li>5. Implement control structures like if-else statements, switch statements, and loops to control the flow of program execution and make decisions.</li><li>6. Define functions in C++ with proper parameter types, return types, and function calls.</li><li>7. Utilize functions to modularize code, enhance code reusability, and perform</li></ol>

	<p>specific tasks.</p> <ol style="list-style-type: none"> <li>8. Understand the concept of arrays in C++ and the process of declaration, initialization, and accessing array elements.</li> <li>9. Work with one-dimensional and multi-dimensional arrays to store and process collections of data.</li> <li>10. Apply problem-solving and algorithmic thinking skills to develop simple programs using C++.</li> <li>11. Demonstrate the ability to debug and troubleshoot common errors in C++ code.</li> <li>12. Collaborate effectively in team-based programming projects, communicating ideas and solutions clearly.</li> </ol>
<p><b>Indicative Contents</b> المحتويات الإرشادية</p>	<p>Indicative content includes the following.</p> <p><u>Theory Part A &amp; B class</u> Learning during attendance class for 15 weeks. [30 hrs] Introduction to C++: Introduce students to the C++ programming language, its syntax, and basic features. [5 hrs]</p> <p>Variables and Data Types: Teach students about variables, their types, and how to declare and use them in C++. Cover fundamental data types such as integers, floating-point numbers, characters, and strings. [5 hrs]</p> <p>Input and Output: Familiarize students with input and output operations in C++, including reading user input and displaying output on the console. [5 hrs]</p> <p>Control Structures: Teach students about control structures in C++, such as if-else statements, switch statements, and loops (while, do-while, for). Explain how these structures can be used to control program flow and make decisions. [5 hrs]</p> <p>Functions: Introduce students to functions in C++, including function definition, parameters, return types, and function calls. Teach them how to create and use functions to modularize their code and perform specific tasks. [5 hrs]</p> <p>Arrays and structure: Cover the concept of arrays in C++, including declaration, initialization, and accessing array elements. Teach students how to work with one-dimensional and multi-dimensional arrays. [5 hrs]</p> <p><u>Practical Part A &amp; B class</u> Learning during the lab practice for 15 weeks. [45 hrs]</p> <p>Code example in C++ programming language, syntax, basic features and how to declare and use them in C++. Cover fundamental data types such as integers floating-point numbers, characters, and strings [15 hrs]</p> <p>Code example in C++ programming language, input and output operations in C++, including reading user input and displaying output on the console as well as control structures in C++, such as if-else statements, switch statements, and loops (while, do-</p>

	<p>while, for). Explain how these structures can be used to control program flow and make decisions. [15 hrs]</p> <p>Code example in C++ programming language, function definition, parameters, return types, and function calls. Teach them how to create and use functions to modularize their code and perform specific tasks and Cover the concept of arrays/ structure in C++, including declaration, initialization, and accessing array elements. Teach students how to work with one-dimensional and multi-dimensional arrays. [15 hrs]</p>
--	---

<b>Learning and Teaching Strategies</b> استراتيجيات التعلم والتعليم	
<b>Strategies</b>	<p>Type something like: 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.</p>

<b>Student Workload (SWL)</b> الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا			
<b>Structured SWL (h/sem)</b> الحمل الدراسي المنتظم للطالب خلال الفصل	73	<b>Structured SWL (h/w)</b> الحمل الدراسي المنتظم للطالب أسبوعيا	5
<b>Unstructured SWL (h/sem)</b> الحمل الدراسي غير المنتظم للطالب خلال الفصل	77	<b>Unstructured SWL (h/w)</b> الحمل الدراسي غير المنتظم للطالب أسبوعيا	5.5
<b>Total SWL (h/sem)</b> الحمل الدراسي الكلي للطالب خلال الفصل	<b>150</b>		

<b>Module Evaluation</b> تقييم المادة الدراسية					
		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
<b>Formative assessment</b>	<b>Quizzes</b>	2	20% (20)	5 and 10	LO #1 - #5
	<b>Assignments</b>	2	20% (20)	2 and 12	LO #1 - #12
	<b>Projects / Lab.</b>				
	<b>Report</b>				

Summative assessment	Midterm Exam	2hr	10% (10)	7	LO #1 - #7
	Final Exam	3hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

<b>Delivery Plan (Weekly Syllabus)</b> المنهاج الاسبوعي النظري	
	Material Covered
Week 1	Introduce students to the C++ programming language.
Week 2	Introduce students to the C++ programming language, its syntax, and basic features.
Week 3	Teach students about variables, their types.
Week 4	Teach students about variables, their types, and how to declare and use them in C++.
Week 5	Teach students about variables, their types, and how to declare and use them in C++. Cover fundamental data types such as integers, floating-point numbers, characters, and strings.
Week 6	Familiarize students with input and output operations in C++, including reading user input and displaying output on the console (1)
Week 7	Familiarize students with input and output operations in C++, including reading user input and displaying output on the console (2)
Week 8	<b>Control</b> Structures: if-else statements, switch statements, and loops (while, do-while, for). Explain how these structures can be used to control program flow and make decisions.
Week 9	<b>Condition</b> Structures: if-else statements, switch statements, and loops (while, do-while, for). Explain how these structures can be used to control program flow and make decisions.
Week 10	Introduce students to functions in C++
Week 11	Learning function definition, parameters, return types, and function calls
Week 12	Teach students how to create and use functions to modularize their code and perform specific tasks.
Week 13	Cover the concept of arrays/ structure in C++, including declaration, initialization, and accessing array elements. Teach students how to work with <b>one-dimensional</b> .
Week 14	Cover the concept of arrays/ structure in C++, including declaration, initialization, and accessing array elements. Teach students how to work with <b>multi-dimensional arrays</b> .
Week 15	Cover the concept of <b>structure</b> in C++, including declaration, initialization, and accessing structure elements.

<b>Week 16</b>	<b>Preparatory week before the final Exam</b>
----------------	---

<b>Delivery Plan (Weekly Lab. Syllabus)</b> المنهاج الاسبوعي للمختبر	
	<b>Material Covered</b>
<b>Week 1</b>	Lab 1: Introduction to C++ language (code 1)
<b>Week 2</b>	Lab 2: The syntax, basic features and how to declare and use them in C++ (code 2)
<b>Week 3</b>	Lab 3: The syntax, basic features and how to declare and use them in C++ (code 3)
<b>Week 4</b>	Lab 4: fundamental data types such as integers floating-point numbers, characters, and strings (code 4)
<b>Week 5</b>	Lab 5: fundamental data types such as integers floating-point numbers, characters, and strings (code 5)
<b>Week 6</b>	Lab 6: The input and output operations in C++, (code 6)
<b>Week 7</b>	Lab 7: input and displaying output on the console (code 7)
<b>Week 8</b>	Lab 8: Using control structures in C++, such as if-else statements (code 8)
<b>Week 9</b>	Lab 9: Using switch statements, (code 9)
<b>Week 10</b>	Lab 10: Using loops (while, do-while, for), (code 10)
<b>Week 11</b>	Lab 11: Explain the flow and make decisions of code (7,8,9, and 10)
<b>Week 12</b>	Lab 12: Using function definition, parameters, return types, and function calls
<b>Week 13</b>	Lab 13: Array one dimension
<b>Week 14</b>	Lab 14: Array two dimensions
<b>Week 15</b>	Lab 15: structure

<b>Learning and Teaching Resources</b> مصادر التعلم والتدريس		
	<b>Text</b>	<b>Available in the Library?</b>
<b>Required Texts</b>	C++ "Beginner level"	Yes
<b>Recommended Texts</b>	Learn C++ Quickly: A Complete Beginner's Guide to Learning C++, Even If You're New to Programming (Crash Course With Hands-On Project)	No
<b>Websites</b>	<a href="https://www.coursera.org/">https://www.coursera.org/</a>	

## 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.