

MODULE DESCRIPTION FORM

Object Oriented Programming

نموذج وصف المادة الدراسية

Module Information				
معلومات المادة الدراسية				
Module Title	Object Oriented Programming		Module Delivery	
Module Type	Core		<input checked="" type="checkbox"/> Theory <input checked="" type="checkbox"/> Lecture <input checked="" type="checkbox"/> Lab <input type="checkbox"/> Tutorial <input type="checkbox"/> Practical <input checked="" type="checkbox"/> Seminar	
Module Code	BCYSCE107-S2			
ECTS Credits	7			
SWL (hr/sem)	175			
Module Level	1	Semester of Delivery		2
Administering Department	CYSCE	College	TECM	
Module Leader	Dr. Zakaria Noor Aldeen Mahmood		e-mail	E-mail
Module Leader's Acad. Title	Lecturer	Module Leader's Qualification	Ph.D.	
Module Tutor		e-mail	zakaria@ntu.edu.iq	
Peer Reviewer Name		e-mail		
Scientific Committee Approval Date	20/06/2023	Version Number	1.0	

Relation with other Modules			
العلاقة مع المواد الدراسية الأخرى			
Prerequisite module	Fundamentals of Programming (BCYSCE102-S1)	Semester	1
Co-requisites module		Semester	

Module Aims, Learning Outcomes and Indicative Contents

أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية

<p>Module Objectives أهداف المادة الدراسية</p>	<ol style="list-style-type: none"> 1. Introduction to OOP and its application. 2. OOP vs Procedure Oriented Programming. 3. Starts from building Classes and member function to advance implementation of OOP programing. 4. Improving the skills of the students through several OOP program implementation and code writing. 5. The students should be able to Identify the OOP uses and purpose of using classes and derived classes, inheritance and polymorphism & the required code lines to perform the needed works 6. as well as qualifying him to use the different kinds of programming style and program functions in building & executing the projects of cyber security engineering.
<p>Module Learning Outcomes مخرجات التعلم للمادة الدراسية</p>	<ol style="list-style-type: none"> 1. Understanding the fundamentals of OOP programming. 2. Mastering OOP programming tools and techniques, including common class inheritance techniques such as multi-level inheritance. 3. Becoming familiar with the OOP concepts such as abstraction, constructor, destructor, <i>this</i> pointer and virtual function. 4. Being competent in common file Input/output with Streams techniques, such as File Stream Class Hierarchy, Opening and Closing files, Read/Write from File. 5. Being able to perform Templates: Function Template, Overloading Function Template, Overloading with Functions. 6. Being able to write Exception Handling: Error Handling, Exception Handling Constructs (try, catch, throw).

Learning and Teaching Strategies

استراتيجيات التعلم والتعليم

<p>Strategies</p>	<p>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>
--------------------------	--

Student Workload (SWL)			
الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا			
Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	79	Structured SWL (h/w) الحمل الدراسي المنتظم للطالب أسبوعيا	5
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	96	Unstructured SWL (h/w) الحمل الدراسي غير المنتظم للطالب أسبوعيا	6
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	175		

Module Evaluation				
تقييم المادة الدراسية				
		Time/Number	Weight (Marks)	Week Due
Formative assessment	Quizzes	4	10% (10)	3,6,9,12
	Assignments	8	5% (5)	Every other week
	Projects / Lab.	14	10% (10)	Continuous
	Report	2	5% (5)	7, 14
	Seminar	1	10% (10)	15
Summative assessment	Midterm Exam	2hr	10% (10)	7
	Final Exam	3hr	50% (50)	15
Total assessment			100% (100 Marks)	

Code BCYSCE 107-S2	Name of the Course Unit	Semester	In-Class Hours (T+P)	Credit	ECTS Credit
	Object Oriented Programming	2	2+3	4	7
GENERAL INFORMATION					
Language of Instruction :		English			
Level of the Course Unit :		BACHELOR'S DEGREE			
Type of the Course :		Compulsory			
Mode of Delivery of the Course Unit		Face to Face			
Coordinator of the Course Unit		Dr. Zakaria Noor Aldeen Mahmood			
Instructor(s) of the Course Unit		Dr. Zakaria Noor Aldeen Mahmood			
OBJECTIVES AND CONTENTS					
Objectives of the Course Unit:		Introducing the fundamentals and principles of OOP programming in C++ language. Starts from building Classes and member function to advance implementation of OOP programing. Improving the skills of the students through several OOP program implementation and code writing. The students should be able to Identify the OOP uses and purpose of using classes and derived classes, inheritance and polymorphism & the required code lines to perform the needed works as well as qualifying him to use the different kinds of programming style and program functions in building & executing the projects of cyber security engineering.			
Contents of the Course Unit:		OOP programming OOP concepts Inheritance types and implementation Runtime Polymorphism Compile-time Polymorphism File Input/output Exception Handling			
	Delivery Plan (Weekly Syllabus)				
	المنهاج الاسبوعي النظري				
WEEK	KEY LEARNING OUTCOMES OF THE COURSE UNIT (On successful completion of this course unit, students/learners will or will be able to)				
1	<ul style="list-style-type: none">Issues with Procedure Oriented Programming				

	Delivery Plan (Weekly Syllabus) المنهاج الاسبوعي النظري
WEEK	KEY LEARNING OUTCOMES OF THE COURSE UNIT (On successful completion of this course unit, students/learners will or will be able to)
2	<ul style="list-style-type: none"> Basic of Object-Oriented Programming (OOP)
3	Procedure Oriented versus Object Oriented Programming
4	Concept of Object-Oriented Programming: Object, Class, Abstraction, Encapsulation, Inheritance, Polymorphism
5	Objects and Classes: C++ Classes, Access Specifiers, Objects and the Member Access, Defining Member Function, Constructor: Default Constructor, Parameterized Constructor, Copy Constructor
6	Destructors, Object as Function Arguments and Return Type, Array of Objects, Pointer to Objects and Member Access, Dynamic Memory Allocation for Objects and Object Array.
7	this Pointer static Data Member and static Function, Constant Member Functions and Constant Objects, Friend Function and Friend Classes
8	Operator Overloading, Overloadable Operators, Syntax of Operator Overloading, Rules of Operator Overloading, Unary Operator Overloading, Binary Operator Overloading, Operator Overloading with Member and Non-Member Functions, Data Conversion: Basic – User Defined and User Defined – User Defined, Explicit Constructors
9	Inheritance: Base and Derived Class, protected Access Specifier, Derived Class Declaration, Member Function Overriding, Forms of Inheritance: single, multiple, multilevel, hierarchical, hybrid, multipath, Multipath Inheritance and Virtual Base Class, Constructor Invocation in Single and Multiple Inheritances, Destructor in Single and Multiple Inheritances
10	Polymorphism and Dynamic Binding: Need of Virtual Function, Pointer to Derived Class, Definition of Virtual Functions, Array of Pointers to Base Class, Pure Virtual functions and Abstract Class, Virtual Destructor, reinterpret cast Operator, Run-Time Type Information, dynamic cast Operator, typed Operator
11	Stream Computation for Console and File Input /Output : Stream Class Hierarchy for Console Input /Output, Testing Stream Errors, Unformatted Input /Output, Formatted Input /Output with iOS Member functions and Flags, Formatting with Manipulators, Stream Operator Overloading
12	File Input/output with Streams, File Stream Class Hierarchy, Opening and Closing files, Read/Write from File, File Access Pointers and their Manipulators, Sequential and Random Access to File, Testing Errors during File Operations
13	<ul style="list-style-type: none"> Templates: Function Template, Overloading Function Template, Overloading with Functions, Overloading with other Template, Class Template, Function Definition of Class Template, On-Template Type, Arguments, Default Arguments with Class Template, Derived Class Template, Introduction to Standard Template Library, Containers, Algorithms, Iterators
14	Exception Handling: Error Handling, Exception Handling Constructs (try, catch, throw), Advantage over Conventional Error Handling, Multiple Exception Handling, Rethrowing Exception, Catching All Exceptions, Exception with Arguments, Exceptions Specification for Function, Handling Uncaught and Unexpected Exceptions.
15	Review/ preparation for final exam
16	Final Exam

Delivery Plan (Weekly Lab. Syllabus) المنهاج الاسبوعي للمختبر	
	Material Covered
Week 1	Lab 1: Getting started - Object-Oriented Programming.
Week 2	Lab 2: Object, Class, Abstraction, Encapsulation, Inheritance, Polymorphism - Examples and Problems
Week 3	Lab 3: Destructors - Examples and Problems.
Week 4	Lab 4: Virtual Destructor
Week 5	Lab 5 :Operator Overloading - Examples and Problems.
Week 6	Lab 6: Inheritance - Examples and Problems.
Week 7	Lab 7: Polymorphism and Dynamic Binding - Examples and Problems.
Week 8	Lab 8: Stream Computation for Console and File Input /Output - Examples and Problems.
Week 9	Lab 9: Templates - Examples and Problems.
Week 10	Lab 10: Exception Handling - Examples and Problems.
Week 11	Lab 11: Debugging - Examples and Problems.
Week 12	Lab 12: Containers, Algorithms, Iterators - Examples and Problems.
Week 13	Lab 13. Conventional Error Handling - Examples and Problems
Week 14	Lab 14: Dynamic Memory - Examples and Problems.
Week 15	Lab 15: review/preparation for final exam
Week 16	Final Exam

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	Available in the Library?
Required Texts	1. Choudhary, H. (2013). C++ Programming-Final Golden Edition. Beginners To Experts Approach Guide-With Easy Learning & Problem Analysis to Program Design & Development.	no

WORKLOAD & ECTS CREDITS OF THE COURSE UNIT Object Oriented Programming			
Workload for Learning & Teaching Activities			
Type of the Learning Activates	Learning Activities (# of week)	Duration (hours, h)	Workload (h)
Lecture & In-Class Activities	15	2	30
Preliminary & Further Study	NA	NA	NA
Land Surveying	NA	NA	NA
Group Work	NA	NA	NA
Laboratory	15	3	45
Reading	6	1	6
Assignment (Homework)	8	2	16
Project Work	NA	NA	NA
Seminar	1	1	1
Internship	NA	NA	NA
Technical Visit	NA	NA	NA
Web Based Learning	6	2	12
Implementation/Application/Practice	NA	NA	NA
Practice at a workplace	NA	NA	NA
Occupational Activity	NA	NA	NA
Social Activity	NA	NA	NA
Thesis Work	NA	NA	NA
Field Study	NA	NA	NA
Report Writing	2	2	4
Final Exam -Theory	1	3	3
Final Exam - Practical	1	1	1
Preparation for the Final Exam- Theory	1	20	20
Preparation for the Final Exam -Practical	1	15	15
Mid-Term Exam - Theory	1	2	2
Mid-Term Exam - Practical	1	1	1
Preparation for the Mid-Term Exam	1	15	15
Short Exam (Quizzes)	4	0.5	2
Preparation for the Short Exam	4	2	8
Total Workload of the Course Unit			175
Recommended Texts	2. Farrell, J. (2008). Object-oriented programming using C++. Cengage Learning.		no

	3. Object-Oriented Programming in C++, Fourth Edition.	
Websites	4. https://www.geeksforgeeks.org/object-oriented-programming-in-cpp/ 5. https://www.w3schools.com/cpp	

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.