



MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information			
معلومات المادة الدراسية			
Module Title	Object Oriented Programming I		Module Delivery
Module Type	Core		☒ Theory ☒ Lecture ☒ Lab ☐ Tutorial ☐ Practical ☐ Seminar
Module Code	CYS201		
ECTS Credits	5		
SWL (hr/sem)	125		
Module Level	2	Semester of Delivery	
Administering Department	Dept. of Cybersecurity	College	College of Information Technology
Module Leader	Dr. Ahmad AlSabhany	e-mail	Ahmad.alsabhany@uofallujah.edu.iq
Module Leader's Acad. Title	Asst. Prof.	Module Leader's Qualification	Ph.D.
Module Tutor	Dr. Senan Ali Abd	e-mail	senan.a.abd@uofallujah.edu.iq
Peer Reviewer Name	Dr. Omar Salah Shareef	e-mail	omar.alshareef@uofallujah.edu.iq
Scientific Committee Approval Date	25-5-2026	Version Number	1.0

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	The module builds upon the programming knowledge and skills acquired in Structured Programming I and II. 1. Develop a solid understanding of object-oriented programming concepts and principles using the Python programming language. 2. Design and implement Python applications utilizing classes, objects, encapsulation, inheritance, polymorphism, and other object-oriented techniques. 3. Apply problem-solving and software design methodologies to create reusable, maintainable, and modular software solutions. 4. Utilize Python libraries and tools for file handling, exception management, and basic data structures in practical programming scenarios. 5. Explore introductory cybersecurity and cryptography concepts through the use of selected Python security and cryptographic libraries. 6. Prepare for advanced object-oriented software development and project-based work in Object-Oriented Programming II through the development of progressively larger Python applications.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	7. Explain the fundamental concepts and principles of object-oriented programming and their implementation in Python. 8. Design and develop Python programs using classes, objects, encapsulation, inheritance, and polymorphism. 9. Apply object-oriented design techniques to create modular, reusable, and maintainable software solutions. 10. Implement file handling, exception handling, and basic data structures within Python applications. 11. Utilize selected Python libraries to perform introductory security and cryptographic operations. 12. Develop medium-scale object-oriented Python applications and demonstrate readiness for advanced project-based development in Object-Oriented Programming II.
Indicative Contents المحتويات الإرشادية	• Introduction to Object-Oriented Programming and Python • Python Development Environment and Program Structure • Variables, Data Types, Operators, and Input/Output • Control Structures and Program Flow • Functions and Modular Programming

	• Python Collections (Lists, Tuples, Dictionaries, and Sets) • Classes and Objects • Constructors and Object Lifecycle • Encapsulation and Data Hiding • Inheritance and Class Hierarchies • Polymorphism and Method Overriding • Special Methods and Operator Overloading • Exception Handling and Error Management • File Handling and Data Persistence • Introduction to Object-Oriented Software Design • Basic Data Structures and Object-Based Applications • Python Standard Libraries for Software Development • Introduction to Python Security and Cryptography Libraries • Basic Hashing, Encoding, and Encryption Applications • Development of Small Object-Oriented Python Projects • Preparation for Advanced Object-Oriented Programming and Software Projects
--	---

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	The module is delivered through lectures, laboratory sessions, practical programming exercises, and guided assignments. Students develop object-oriented programming skills through hands-on Python programming, problem-solving activities, and small software projects. Selected practical activities introduce Python libraries relevant to cybersecurity and cryptography, preparing students for advanced development work in Object-Oriented Programming II.

Student Workload (SWL) الحمل الدراسي للطلاب محسوب لـ ١٥ اسبوعا			
Structured SWL (h/sem) الحمل الدراسي المنتظم للطلاب خلال الفصل	78	Structured SWL (h/w) الحمل الدراسي المنتظم للطلاب أسبوعيا	5
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطلاب خلال الفصل	47	Unstructured SWL (h/w) الحمل الدراسي غير المنتظم للطلاب أسبوعيا	4
Total SWL (h/sem)	125		

الحمل الدراسي الكلي للطلاب خلال الفصل	
---------------------------------------	--

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

Delivery Plan (Weekly Syllabus)	
المنهاج الاسبوعي النظري	
	Material Covered
Week 1	Python Review: Syntax, Data Types, Control Structures, Functions
Week 2	Python Review: Collections, Modules, and Program Organization
Week 3	Introduction to Object-Oriented Programming, Classes and Objects
Week 4	Constructors, Attributes, and Methods
Week 5	Encapsulation and Data Hiding
Week 6	Inheritance and Class Hierarchies
Week 7	Polymorphism and Method Overriding
Week 8	Midterm Examination
Week 9	Special Methods and Operator Overloading
Week 10	Exception Handling and File Operations
Week 11	Object Relationships and Advanced Class Design
Week 12	Object-Oriented Design and Small Applications
Week 13	Python Security Libraries and Hashing
Week 14	Python Cryptography Libraries and Secure Programming Basics
Week 15	Mini Project Development, Review, and Final Preparation

Week 16	Final Exam Preparation
---------	------------------------

Delivery Plan (Weekly Lab. Syllabus) المنهاج الاسبوعي للمختبر	
	Material Covered
Week 1	Python Environment Setup and Review Exercises
Week 2	Functions, Collections, and Modular Programming Exercises
Week 3	Creating and Using Classes and Objects
Week 4	Constructors, Attributes, and Methods
Week 5	Encapsulation and Access Control Techniques
Week 6	Implementing Inheritance in Python
Week 7	Polymorphism and Method Overriding Applications
Week 8	Midterm Practical Assessment
Week 9	Special Methods and Operator Overloading Exercises
Week 10	Exception Handling and File Processing Applications
Week 11	Designing Multi-Class Applications
Week 12	Object-Oriented Software Design Case Study
Week 13	Python Hashing and Security Library Applications
Week 14	Cryptography Library Applications (Hashing, Encryption, and Decryption)
Week 15	Mini Project Implementation and Demonstration

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	Available in the Library?
Required Texts	1. Python Crash Course, 3rd Edition, Eric Matthes. 2. Automate The Boring Stuff with Python, Ed Edition, Al Sweigart.	Yes
Recommended Texts	3. Python 3 Object-Oriented Programming, Third Edition, Dusty Phillips.	Yes
Websites	1. https://github.com/ehmatthes/pcc_3e 2. https://www.w3schools.com/python/	

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.				