University of Fallujah
College of Information Technology
Department of Cybersecurity

MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information معلومات المادة الدراسية					
Module Title	**Object Oriented Programming II**			**Module Delivery**	
Module Type	**Core**			☒ **Theory** ☒ **Lecture** ☒ **Lab** ☐ **Tutorial** ☐ **Practical** ☐ **Seminar**	
Module Code	**CYS211**				
ECTS Credits	**5**				
SWL (hr/sem)	**125**				
Module Level	2		**Semester of Delivery**	2	
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, and object-oriented programming I. 1. Develop an advanced understanding of object-oriented programming principles and software design methodologies using the Python programming language. 2. Design, implement, and maintain medium-scale software systems through the effective application of object-oriented concepts, including abstraction, encapsulation, inheritance, polymorphism, and object relationships. 3. Apply software engineering practices to develop modular, reusable, scalable, and maintainable Python applications. 4. Utilize advanced Python libraries and frameworks to support software development, data processing, and security-related applications. 5. Develop practical skills in the use of Python cryptography and security libraries for implementing encryption, hashing, authentication, and secure programming techniques. 6. Gain hands-on experience in planning, developing, testing, documenting, and presenting intermediate-level object-oriented software projects. 7. Prepare students for advanced cybersecurity, software development, and project-based modules through the completion of a substantial Python application.
Module Learning Outcomes **مخرجات التعلم للمادة الدراسية**	1. Analyze and apply advanced object-oriented programming concepts and software design principles in Python applications. 2. Design and develop medium-scale object-oriented software systems using appropriate class structures, object relationships, and design techniques. 3. Implement modular, reusable, scalable, and maintainable Python applications following established software engineering practices. 4. Utilize advanced Python libraries and frameworks to support software development and security-related applications. 5. Apply Python cryptography and security libraries to implement encryption, hashing, authentication, and secure programming solutions. 6. Debug, test, document, and evaluate object-oriented software applications using appropriate development methodologies and tools. 7. Plan, develop, and present an intermediate-level software project that demonstrates proficiency in object-oriented programming, software design, and secure application development.

Indicative Contents **المحتويات الإرشادية**	• Advanced Class Design and Object Relationships • Composition and Aggregation • Abstract Classes and Interfaces • Advanced Inheritance and Polymorphism • Operator Overloading and Special Methods • Exception Handling and Custom Exceptions • File Processing and Data Persistence • Object-Oriented Software Design Principles • Introduction to Design Patterns • Modular Application Development • Python Packages and Third-Party Libraries • Testing, Debugging, and Code Documentation • Secure Programming Practices • Python Cryptography Libraries • Hashing, Encryption, and Authentication Applications • Development of Multi-Class Software Systems • Intermediate-Level Object-Oriented Project Design • Software Project Implementation and Documentation • Project Presentation and Evaluation

Learning and Teaching Strategies **استراتيجيات التعلم والتعليم**	
Strategies	**The module is delivered through lectures, laboratory sessions, project-based learning, and guided practical exercises. Students strengthen their object-oriented programming skills through the design and development of medium-scale Python applications. Emphasis is placed on software design, secure programming practices, the use of Python cryptography libraries, teamwork, problem-solving, and the implementation of an intermediate-level software project.**

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

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 - #7
	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	**Review of OOP Concepts and Advanced Class Design**
Week 2	**Composition, Aggregation, and Object Relationships**
Week 3	**Abstract Classes and Interfaces**
Week 4	**Advanced Inheritance and Polymorphism**
Week 5	**Operator Overloading and Special Methods**
Week 6	**Exception Handling and Custom Exceptions**
Week 7	**File Processing and Data Persistence**
Week 8	**Midterm Examination**
Week 9	**Object-Oriented Software Design Principles**
Week 10	**Introduction to Design Patterns**
Week 11	**Modular Application Development and Python Packages**

Week 12	**Testing, Debugging, and Code Documentation**
Week 13	**Secure Programming Practices and Python Security Libraries**
Week 14	**Python Cryptography Libraries and Authentication Applications**
Week 15	**Intermediate Project Development, Review, and Final Preparation**
Week 16	**Final Exam Preparation**

Delivery Plan (Weekly Lab. Syllabus) **المنهاج الاسبوعي للمختبر**	
	Material Covered
Week 1	**Practical Topics**
Week 2	**OOP Review and Advanced Class Design Exercises**
Week 3	**Composition, Aggregation, and Object Relationship Applications**
Week 4	**Implementing Abstract Classes and Interfaces**
Week 5	**Advanced Inheritance and Polymorphism Applications**
Week 6	**Operator Overloading and Special Methods**
Week 7	**Custom Exceptions and Error Handling Applications**
Week 8	**File Processing and Data Persistence Projects**
Week 9	**Midterm Practical Assessment**
Week 10	**Object-Oriented Software Design Case Study**
Week 11	**Design Pattern Implementation Exercises**
Week 12	**Modular Application Development Using Packages and Modules**
Week 13	**Testing, Debugging, and Documentation of Python Applications**
Week 14	**Secure Programming and Python Security Library Applications**
Week 15	**Cryptography Library Applications (Encryption, Hashing, and Authentication)**

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.