



MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information			
معلومات المادة الدراسية			
Module Title	Data Structures		Module Delivery
Module Type	Core		☒ Theory ☒ Lecture ☒ Lab ☐ Tutorial ☐ Practical ☐ Seminar
Module Code	CYS205		
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. Mohammed Abduljawad M. Al-Shibly	e-mail	mohammed.abduljawad@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 fundamental data structures and their role in efficient software development and problem solving. 2. Design and implement core data structures in C++, including arrays, linked lists, stacks, queues, trees, and related structures. 3. Apply pointers, dynamic memory allocation, and object-oriented programming techniques in the implementation of data structures. 4. Analyze the performance and efficiency of data structures with respect to memory usage and computational complexity. 5. Select appropriate data structures to solve computational and cybersecurity-related problems. 6. Develop practical programming skills through the implementation, testing, and evaluation of data structure applications in C++. 7. Introduce essential Python data structures, including tuples, dictionaries, and sets, and compare their use with equivalent C++ structures. 8. Prepare students for advanced study in algorithms, secure software development, artificial intelligence, and cybersecurity applications.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	1. Explain the characteristics, operations, and applications of fundamental data structures. 2. Design and implement data structures in C++ using arrays, linked lists, stacks, queues, trees, and related structures. 3. Apply pointers and dynamic memory allocation techniques in the development of efficient data structure implementations. 4. Analyze the efficiency of data structures using basic complexity and performance measures. 5. Select and utilize appropriate data structures to solve computational problems effectively. 6. Develop, test, and debug data structure applications using the C++ programming language. 7. Utilize essential Python data structures, including tuples, dictionaries, and sets, and compare their use with equivalent C++ structures. 8. Demonstrate readiness for advanced study in algorithms, software development, artificial intelligence, and cybersecurity applications.
Indicative Contents المحتويات الإرشادية	● Introduction to Data Structures and Abstract Data Types (ADTs) ● Arrays and Memory Representation ● Pointers and Dynamic Memory Allocation ● Recursive Problem Solving

	• Linked Lists (Singly, Doubly, and Circular) • Stacks and Their Applications • Queues, Circular Queues, and Priority Queues • Trees and Binary Trees • Binary Search Trees • Tree Traversal Techniques • Heaps and Priority Queue Implementations • Hashing and Hash Tables • Graph Fundamentals and Representations • Searching Techniques • Introduction to Sorting Techniques • Complexity Analysis and Performance Evaluation • Data Structure Selection and Optimization • Introduction to C++ Standard Template Library (STL) • Python Data Structures: Lists, Tuples, Dictionaries, and Sets • Comparative Analysis of Data Structures in C++ and Python • Applications of Data Structures in Software Development and Cybersecurity
--	--

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	The module is delivered through lectures, laboratory sessions, practical programming exercises, and problem-solving activities. Students develop data structure implementation skills through hands-on programming in C++ and selected Python data structures. Real-world examples and case studies are used to demonstrate how data structures support efficient software systems, cryptographic algorithms, and cybersecurity applications. Laboratory activities emphasize the design, implementation, testing, and performance evaluation of data structures in practical computing and security-related scenarios.

Student Workload (SWL) الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا			
Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	63	Structured SWL (h/w) الحمل الدراسي المنتظم للطالب أسبوعيا	4
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	62	Unstructured SWL (h/w) الحمل الدراسي غير المنتظم للطالب أسبوعيا	4

Total SWL (h/sem) الحمل الدراسي الكلي للطلاب 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
	Practical (Lab)	1	10% (10)	9	LO #1 - #4
	Quiz	1	10% (10)	13	All
Summative assessment	Midterm Exam	2hr	10% (10)	7	LO #5 - #8
	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. Drozdek, A. (2013). Data Structures and Algorithms in C++ (4th ed.). Cengage Learning. 2. Carrano, F. M., & Henry, T. M. (2025). <i>Secure Data Structures and Algorithms with C++: Walls and Mirrors</i> (8th ed.). Pearson.	Yes
Recommended Texts	1. Goodrich, M. T., Tamassia, R., & Mount, D. M. (2014). Data Structures and Algorithms in C++ (2nd ed.). Wiley. 2. Ranum, D. L., & Miller, B. N. (2011). Problem Solving with Algorithms and Data Structures Using Python. Franklin, Beedle & Associates.	Yes
Websites	1. https://www.w3schools.com/cpp/default.asp	

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.				