



MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية



Module Information			
معلومات المادة الدراسية			
Module Title	Data Structure		Module Delivery
Module Type	Core		☒ Theory ☐ Lecture ☒ Lab ☐ Tutorial ☒ Practical ☒ Seminar
Module Code	Cys2112		
ECTS Credits	8		
SWL (hr/sem)	200		
Module Level	UGII	Semester of Delivery	
Administering Department	Cybersecurity	College	College of Computer Science & Information Technology
Module Leader	Ali Kareem Abdul Raheem	e-mail	alialmujab@uowa.edu.iq
Module Leader's Acad. Title	Lecturer	Module Leader's Qualification	Ph.D.
Module Tutor	Ali Kareem Abdul Raheem	e-mail	alialmujab@uowa.edu.iq
Peer Reviewer Name	Dr. Haider Mohammed Ali	e-mail	hayder.alghanami@uowa.edu.iq
Scientific Committee Approval Date	21/9/2026	Version Number	1.0

Relation with other Modules			
العلاقة مع المواد الدراسية الأخرى			
Prerequisite module	CyS1213	Semester	2
Co-requisites module	None	Semester	...


 د. م. علي كريم عبد الرحمن
 ر. ق. الامن الميسري
 ٢٠٢٦ - ٢٠٢٧




 ٢٠٢٦ - ٢٠٢٧

Department Head Approval

Dean of the College Approval

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Objectives أهداف المادة الدراسية	<p>Comprehend the fundamental principles of data structures and their significance in software development.</p> <p>Classify various types of data structures and understand the core operational logic of each.</p> <p>Identify practical applications of data structures across computer science and related scientific fields.</p>
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	<ol style="list-style-type: none"> 1. Mastery of Fundamental Data Structures: Gain a comprehensive understanding of core structures, including arrays, linked lists, stacks, queues, trees, graphs, and hash tables, focusing on their unique characteristics and operational logic. 2. Practical Implementation: Develop proficiency in implementing and manipulating various data structures using high-level programming languages, bridging the gap between theoretical concepts and hands-on application. 3. Algorithmic Problem Solving: Enhance the ability to model real-world challenges by selecting and designing appropriate data structures to create robust, scalable solutions. 4. Memory Management & Optimization: Acquire essential knowledge of dynamic memory allocation, pointer manipulation, and optimization techniques to ensure efficient resource management and prevent memory leaks. 5. Algorithmic Efficiency: Cultivate advanced algorithmic thinking by analyzing time and space complexity, ensuring that designed solutions are optimized for maximum computational performance.
Indicative Contents المحتويات الإرشادية	<p>This course explores the fundamental components of data structures, emphasizing both theoretical foundations and practical implementation. The curriculum is designed to provide students with the tools necessary to model data efficiently and solve complex computational problems.</p>

	<p>Part A: Foundational Principles</p> <p>Introduction to Data Structures [10 hrs] Overview of data structures and their critical role in problem-solving.</p> <p>Introduction to basic terminology, Abstract Data Types (ADTs), and collection concepts.</p> <p>Arrays [15 hrs] Study of arrays as a fundamental, contiguous data structure.</p> <p>Understanding indexing, memory mapping, and element manipulation.</p> <p>Exploring multi-dimensional arrays and their practical applications.</p> <p>Part B: Linear Data Structures</p> <p>Stacks [15 hrs] Analysis of the LIFO (Last-In-First-Out) behavior and logic.</p> <p>Implementation of core operations: Push, Pop, and Peek. Applications in expression evaluation (infix/postfix), function call management, and backtracking algorithms.</p> <p>Queues [15 hrs] Introduction to FIFO (First-In-First-Out) mechanisms and operations (Enqueue / Dequeue).</p> <p>Examination of specialized types: Circular queues, Priority queues, and Double-Ended Queues (Dequeues).</p> <p>Real-world applications in scheduling and buffer management.</p> <p>Part C: Non-Linear Structures & Algorithmic Analysis</p> <p>Linked Lists [15 hrs] Introduction to dynamic data structures and pointer-based memory management.</p> <p>Implementation of Singly, Doubly, and Circular Linked Lists.</p> <p>Core operations: Insertion, deletion, traversal, and searching.</p> <p>Trees [15 hrs] Study of hierarchical structures: Binary Trees, Binary Search Trees (BST), AVL Trees, and B-Trees.</p> <p>Mastery of tree traversal algorithms: Pre-order, In-order, and Post-order.</p> <p>Operational logic for efficient searching, insertion, and deletion.</p>
--	--

	<p>Searching Algorithms [15 hrs]</p> <p>Detailed analysis of search methodologies, focusing on Linear Search and Binary Search efficiency.</p> <p>Graph Algorithms [15 hrs]</p> <p>Representation of graphs and implementation of traversal techniques: Breadth-First Search (BFS) and Depth-First Search (DFS).</p> <p>Hash Tables [15 hrs]</p> <p>Understanding hashing principles, hash functions, and collision resolution techniques (Chaining and Open Addressing).</p> <p>Implementing efficient lookups, dictionary structures, and data indexing.</p> <p>Recursion [15 hrs]</p> <p>Theoretical and practical exploration of recursive logic and its application in optimizing data structure traversals and algorithms.</p>
--	--

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	<p>Throughout the module, effective learning and teaching strategies will be employed to ensure that students understand the concepts and develop the necessary skills. The main strategies adopted in delivering this module include:</p> <p>Delivering theoretical lectures.</p> <p>Conducting practical exercises in laboratories to provide students with hands-on experience and improve their programming skills.</p> <p>Conducting theoretical examinations to evaluate students' understanding of the course material.</p> <p>Assigning reports to students to develop their research and academic writing skills on specific topics.</p> <p>Assigning small projects to student groups using real-world examples and case studies to demonstrate the importance and applications of Object-Oriented Programming.</p>

Student Workload (SWL)			
الحمل الدراسي للطلاب محسوب لـ ١٥ اسبوعا			
Structured SWL (h/sem) الحمل الدراسي المنتظم للطلاب خلال الفصل	63	Structured SWL (h/w) الحمل الدراسي المنتظم للطلاب أسبوعيا	4
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطلاب خلال الفصل	137	Unstructured SWL (h/w) الحمل الدراسي غير المنتظم للطلاب أسبوعيا	9
Total SWL (h/sem) الحمل الدراسي الكلي للطلاب خلال الفصل	200		

Module Evaluation					
تقييم المادة الدراسية					
		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	Quizzes	2	10% (10)	5 and 10	CLO1, CLO5
	Assignments	1	5% (5)	2 and 12	CLO1, CLO2, CLO3
	Projects / Lab.	1	10% (10)	Continuous	CLO2, CLO4, CLO5
	Projects / Pr.	2	10% (10)	6 and 15	CLO2, CLO3, CLO4, CLO5
	Seminar	1	5% (5)	14	CLO1, CLO3, CLO5
Summative assessment	Midterm Exam	2hr	10% (10)	8	CLO1, CLO2, CLO5
	Final Exam	3hr	50% (50)	17	CLO1, CLO2, CLO3, CLO4, CLO5
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)	
المنهاج الاسبوعي النظري	
	Material Covered
Week 1	Introduction to Data structures
Week 2	Arrays, structures, pointers, memory allocation

Week 3	Stack and their basic operations.
Week 4	Queue and their basic operations.
Week 5	Recursion
Week 6	Single linked list
Week 7	double linked list
Week 8	circular linked list
Week 9	Midterm Exam
Week 10	Trees, binary search trees and basic operations
Week 11	Graphs and basic algorithms on graphs: depth first and breadth first search
Week 12	Sorting Algorithms
Week 13	Searching Algorithms
Week 14	Dijkstra's algorithm. Priority queues
Week 15	Seminar Discussion

Delivery Plan (Weekly Lab. Syllabus) المنهاج الاسبوعي للمختبر	
	Material Covered
Week 1	Lab 1: Programming fundamentals overview
Week 2	Lab 2: Arrays, structures, pointers, memory allocation
Week 3	Lab 3: Stack and their basic operations.
Week 4	Lab 4: Queue and their basic operations.
Week 5	Lab 5: Recursion
Week 6	Lab 6: Single linked list
Week 7	Lab 7: double linked list
Week 8	midterm exam
Week 9	Lab 9: circular linked list
Week 10	Lab 10: Trees, binary search trees and basic operations
Week 11	Lab 12: Graphs and basic algorithms on graphs: depth first and breadth first search

Week 12	Lab 14: Sorting Algorithms
Week 13	Lab 15: Searching Algorithms
Week 14	Lab 13: Dijkstra's algorithm. Priority queues
Week 15	Project Submission

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	Available in the Library?
Required Texts	✓ Clifford A. Shaffer (2009). A Practical Introduction to Data Structures and Algorithm Analysis(3rd) , Prentice Hall ✓ MICHAEL MCMILLAN.(2007). DATA STRUCTURES AND ALGORITHMS USING C#. Michael McMillan 2007	
Recommended Texts		
Websites	https://www.w3schools.com/dsa/dsa_intro.php	

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.