MODULE DESCRIPTION FORM

Module Information				
Module Title	**Data Structure**		**Module Delivery**	
Module Type	**Core**		**Lecture** **Lab** **Practical**	
Module Code	**IT2204**			
ECTS Credits	**6**			
SWL (hr/sem)	**150**			
Module Level	UG2	**Semester of Delivery**	2	
Administering Department	Information Technology	**College**	College of computer science and information technology	
Module Leader	Karar Sadiq Mohsen Al-Ghadhri	**e-mail**	karar.sadeq@uowa.edu.iq	
Module Leader's Acad. Title	Lecturer	**Module Leader's Qualification**	M.Sc.	
Module Tutor	Karar Sadiq Mohsen Al-Ghadhri	**e-mail**	karar.sadeq@uowa.edu.iq	
Peer Reviewer Name	Asst. Prof Haider Mohammed	**e-mail**	hayder.alghanami@uowa.edu.iq	
Scientific Committee Approval Date	2026-01-10	**Version Number**	1	

Relation with other Modules			
Prerequisite module	Programming Fundamentals 2	**Semester**	2
Co-requisites module	-	**Semester**	-

جامعة وارث الانبياء
كلية العلوم
قسم تكنولوجيا المعلومات

جامعة وارث الانبياء
عمادة كلية العلوم

2026 / 2025

Department Head Approval **Dean of the College Approval**

Module Aims, Learning Outcomes and Indicative Contents	
Module Objectives	1. Understand the general principle of data structures and their usefulness 2. Knowing the types of data structures and the main principle of each structure 3. Knowledge of applications related to the subject and other fields of knowledge
Module Learning Outcomes	1. Understanding Fundamental Data Structures: The course aims to provide students with a solid understanding of fundamental data structures such as arrays, linked lists, stacks, queues, trees, graphs, and hash tables. Students learn about the characteristics, operations, and implementation details of these data structures. 2. Implementing Data Structures: The course focuses on teaching students how to implement various data structures in a programming language of their choice. This hands-on experience helps students gain practical skills in implementing and manipulating data structures efficiently. 3. Problem Solving: Data Structures course typically includes problem-solving exercises and assignments. Students are presented with real-world problems and learn how to model them using appropriate data structures. They develop problem-solving skills by designing algorithms and implementing solutions using the learned data structures.
	4. Memory Management: Understanding data structures also involves understanding how memory is managed in a program. Students learn about concepts such as dynamic memory allocation, pointers, memory leaks, and memory optimization techniques to ensure efficient use of memory in data structures. 5. Algorithmic Thinking and Efficiency: The course cultivates algorithmic thinking skills in students. They learn how to break down complex problems into smaller, solvable tasks and design efficient algorithms using appropriate data structures. Emphasis is placed on optimizing algorithms for time and space efficiency.

Indicative Contents	In Data Structures, there are several key components that should be considered. While the specific contents may vary depending on the nature of the structure and the context in which it is being used, the basic contents in Data Structures are: <u>Part A – Data structures principles</u> 1. Introduction to Data Structures: Overview of data structures, their importance, and their role in problem-solving. Introduction to the basic terminology and concepts related to data structures (collections) [10 hrs] 2. Arrays: Study of arrays as a fundamental data structure. Understanding the concept of indexing, accessing, and manipulating array elements. Exploring multidimensional arrays and their applications. [15 hrs] <u>Part B – Linear Structures</u> 1. Stacks: Understanding the stack data structure and its LIFO (Last-In-First-Out) behavior. Implementing stack operations, such as push, pop, and peek. Applications of stacks, such as expression evaluation, function call management, and backtracking . [15 hrs] 2. Queues: Introduction to queues and their FIFO (First-In-First-Out) behavior. Implementing queue operations, such as enqueue and dequeue. Exploring different types of queues, including circular queues and priority queues.

Applications of queues in real-world scenarios
[15 hrs]

Part C – Nonlinear Structures

1. Linked Lists: Introduction to linked lists as a dynamic data structure. Understanding the different types of linked lists, such as singly linked lists, doubly linked lists, and circular linked lists. Operations on linked lists, including insertion, deletion, traversal, and searching. [15 hrs]
2. Trees: Study of tree structures, including binary trees, binary search trees, AVL trees, and B-trees. Understanding tree traversal algorithms (pre-order, in-order, post-order) and operations such as insertion, deletion, and searching in trees. Applications of trees, such as hierarchical data representation and sorting [15 hrs]
3. searching algorithms (e.g., binary search) [15 hrs]
4. graph algorithms (e.g., breadth-first search, depth-first search) [15 hrs]
5. Hash Tables: Understanding the concept of hashing and hash functions. Exploring hash table data structure and collision resolution techniques (chaining, open addressing). Implementing hash table operations, such as insertion, deletion, and searching. Applications of hash tables, such as dictionary implementations and efficient lookup. [15 hrs]
6. Recursion: Understanding the concept of recursion and its applications in data structures and algorithms. [15 hrs]

Learning and Teaching Strategies	
Strategies	Throughout the module, it is important to employ effective learning and teaching strategies to ensure students grasp the concepts and develop the necessary skills. The main strategy that will be adopted in delivering this module: Giving lectures. Practical Exercises within the laboratories: Provide students with opportunities to practice their skills through practical exercises. Conducting theoretical exams. Request a report from students to teach them how to do research on a specific topic. Request some small projects from student groups of real-world examples and case studies to demonstrate the relevance and impact of different Data structures

Student Workload (SWL)			
Structured SWL (h/sem)	60	**Structured SWL (h/w)**	4
Unstructured SWL (h/sem)	87	**Unstructured SWL (h/w)**	5
Total SWL (h/sem)	147 + 3 final = 150		

Module Evaluation

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	**Quizzes**	5	10% (8)	All Weeks	5,10
	Onsite Assignments	5	10% (7)	All Weeks	6
	Lab	5	10% (15)	All Weeks	1,2,3,4,5,6,7,8,9,10
	Home Work	5	10% (7)	All Weeks	6
	Project	1	10% (5)	14	6,15
Summative assessment	**Midterm Exam**	2hr	10% (10)	8	
	Final Exam	3hr	50% (50)	17	
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)

	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	circular linked list 2
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.
Week 15	Priority queues
Week 16	**Preparatory week before the final Exam**

Delivery Plan (Weekly Lab. Syllabus)	
	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	**Lab 8: double linked list 2**

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

	Text	**Available in the Library?**
Required Texts	1. Clifford A. Shaffer (2009). A Practical Introduction to Data Structures and Algorithm Analysis (3rd), Prentice Hall 2. MICHAEL MCMILLAN. (2007). DATA STRUCTURES AND ALGORITHMS USING C#. Michael McMillan 2007 3. Granville Barnett, and Luca Del Tongo (2008). Data Structures and Algorithms: Annotated Reference with Examples (1st)	No
Recommended Texts		
Websites		

Grading Scheme				
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.