MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information معلومات المادة الدراسية					
Module Title	**Data Structures هياكل البيانات**			**Module Delivery**	
Module Type	**Core**			☒ **Theory** ☒ **Lecture** ☒ **Lab** ☐ **Tutorial** ☐ **Practical** ☐ **Seminar**	
Module Code	CYBT206				
ECTS Credits	**4**				
SWL (hr/sem)	**100**				
Module Level			**Semester of Delivery**	4	
Administering Department	Department of Cybersecurity Technical Engineering		**College**	Technical Engineering College of Computer and Artificial Intelligence - Kirkuk	
Module Leader			**e-mail**		
Module Leader's Acad. Title			**Module Leader's Qualification**		
Module Tutor			**e-mail**		
Peer Reviewer Name			**e-mail**		
Scientific Committee Approval Date	02/11/2026		**Version Number**	1.0	

Relation with other Modules العلاقة مع المواد الدراسية الأخرى			
Prerequisite module	C++	**Semester**	1
Co-requisites module	Database Systems	**Semester**	3

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Aims أهداف المادة الدراسية	This module aims to introduce students to the fundamental concepts, design, and implementation of data structures and algorithms. It focuses on abstract data types (ADTs), memory representation, array-based and pointer-based implementations, linear data structures (stacks, queues, linked lists), non-linear data structures (trees, binary search trees, heaps, graphs), sorting and searching algorithms, and basic algorithmic complexity analysis (Big-O notation).
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	Understand abstract data types (ADTs), memory allocation (static vs dynamic), and pointers in C/C++. Analyze time and space complexity of algorithms using asymptotic notation (Big-O, Big-Omega, Big-Theta). Implement and manipulate linear data structures (arrays, stacks, queues, singly/doubly/circular linked lists). Master non-linear data structures (binary trees, binary search trees (BST), AVL trees, heaps, priority queues). Understand and implement graph representation techniques (adjacency matrix/list) and graph traversal algorithms (BFS & DFS). Implement and evaluate fundamental searching (linear, binary search) and sorting algorithms (bubble, insertion, selection, quicksort, mergesort). Solve complex computational problems by selecting appropriate data structures for optimal time/space efficiency.
Indicative Contents المحتويات الإرشادية	The module will cover the following key areas: Introduction to Data Structures, Abstract Data Types (ADTs), Memory Management & Pointers in C/C++ Algorithmic Complexity Analysis: Time/Space Trade-offs, Asymptotic Notations (Big-O, Big-Ω, Big-Θ) Arrays, Vectors, and Array-Based List Implementations Stacks: Array and Linked List Implementations, Applications (Infix/Postfix conversion, Function Call Stack) Queues: Standard Queue, Circular Queue, Double-Ended Queue (Deque), and Priority Queues Linked Lists: Singly Linked Lists, Doubly Linked Lists, Circular Linked Lists, and Operations Trees: Tree Terminology, Binary Trees, Tree Traversals (In-order, Pre-order, Post-order) Binary Search Trees (BST): Insertion, Deletion, Search Operations, and Self-Balancing Trees (AVL Trees) Heaps & Priority Queues: Max/Min Heaps, Heapify Operation, and Heapsort Graph Data Structures: Terminology, Adjacency Matrix & List Representations, Graph Traversals (BFS & DFS) Searching & Hashing: Linear Search, Binary Search, Hash Tables, Hash Functions, and Collision Resolution Sorting Algorithms: Comparison-Based Sorts (Bubble, Selection, Insertion) and Divide-and-Conquer Sorts (Quicksort, Mergesort)

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	The module will employ a combination of strategies including: Lectures: To provide theoretical foundations on data structures, ADTs, and algorithm complexity analysis. Group Discussions: To evaluate algorithm efficiency and choose optimal data structures for given problem scenarios. Problem-Solving Sessions: To trace algorithms, analyze time complexity equations, and draw tree/graph structures. Practical Lab Sessions: Hands-on experience implementing data structures and algorithms in C++ / Java / Python. Tutorials: To practice complex pointer manipulation, tree balancing, and recursion-to-iteration transformations. Mid-term and final exams: To assess comprehensive theoretical knowledge and coding problem-solving skills. Preparatory Week: To give students time to revise and clarify any difficulties before the final exam.

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	2
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	37	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	100		

Module Evaluation تقييم المادة الدراسية					
		Time/Number	**Weight (Marks)**	**Week Due**	**Relevant Learning Outcome**
Formative assessment	**Quizzes**	2	5% (5)	5, 10	LO #1, 3 and 5
	Assignments	2	5% (5)	2, 12	LO #2, 4 and 6
	Projects / Lab.	1	10% (10)	Continuous	LO #7
	Report	1	10% (10)	13	LO #3, 5 and 7
	Midterm Exam	2 hr	20% (20)	8	LO #1 - 4

Summative assessment	**Final Exam**	2hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus) المنهاج الاسبوعي النظري	
	Material Covered
Week 1	Introduction to Data Structures, ADTs, Memory Allocation, and Pointers in C/C++.
Week 2	Algorithm Analysis: Time & Space Complexity, Asymptotic Notation (Big-O, Big-Omega, Big-Theta).
Week 3	Stacks: ADT Definition, Array Implementation, Stack Applications (Infix to Postfix Conversion).
Week 4	Queues: Linear Queues, Circular Queues, Deque, and Queue Applications.
Week 5	Linked Lists I: Singly Linked Lists Operations (Node Creation, Insertion, Deletion, Traversal).
Week 6	Linked Lists II: Doubly Linked Lists, Circular Linked Lists, Stack & Queue via Linked Lists.
Week 7	Recursion: Principles, Call Stack Behavior, Recursion vs Iteration, and Backtracking.
Week 8	Mid-term Exam (الفصل منتصف اختبار) & Binary Trees: Terminology, Properties, Tree Traversals (Inorder, Preorder, Postorder).
Week 9	Binary Search Trees (BST): Definition, Searching, Insertion, Deletion Algorithms, and BST Complexity.
Week 10	Balanced Search Trees: Introduction to AVL Trees, Rotations (LL, RR, LR, RL), and Height Balancing.
Week 11	Heaps & Priority Queues: Max/Min Heap Properties, Heapify Operation, Priority Queues, and Heapsort.
Week 12	Graph Fundamentals: Terminology, Directed/Undirected Graphs, Adjacency Matrix and Adjacency List Representations.
Week 13	Graph Algorithms: Breadth-First Search (BFS) and Depth-First Search (DFS) Traversals.
Week 14	Searching & Hashing: Linear & Binary Search, Hash Tables, Hash Functions, Collision Resolution (Chaining, Open Addressing).
Week 15	Sorting Algorithms: Comparison Sorts (Bubble, Selection, Insertion) vs Divide-and-Conquer Sorts (Quicksort, Mergesort).
Week 16	**Final Review, Algorithm Efficiency Comparison & Final Examination (النهائي الاختبار).**

Delivery Plan (Weekly Lab. Syllabus) المنهاج الاسبوعي للمختبر	
	Material Covered
Week 1	Lab 1: Pointers, Dynamic Memory Allocation (malloc/new), and Structures/Classes in C++.
Week 2	Lab 2: Array-Based Implementation of Stacks and Infix-to-Postfix Expression Evaluator.
Week 3	Lab 3: Array-Based Implementation of Linear and Circular Queues.
Week 4	Lab 4: C++ Implementation of Singly and Doubly Linked Lists (Node Management Operations).
Week 5	Lab 5: C++ Implementation of Binary Search Tree (BST) Operations and Recursive Traversals.
Week 6	Lab 6: Implementation of Graph Traversals (BFS & DFS) and Sorting Algorithms (Quicksort & Mergesort).
Week 7	Lab 7: Revision, Data Structures Course Project Evaluation, Practical Exam, and Course Assessment.

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	**Available in the Library?**
Required Texts	Weiss, M. A. (2014). "Data Structures and Algorithm Analysis in C++" (4th ed.). Pearson.	Yes
Recommended Texts	Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). "Introduction to Algorithms" (4th ed.). MIT Press.	Yes
Websites	Yes	

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.