MODULE DESCRIPTION FORM

Module Information					
Module Title	**Data Structures 2**			**Module Delivery**	
Module Type	**Basic**			☑ **Theory** ☑ **Lecture** ☑ **Lab** ☑ **Tutorial** ☑ **Practical** ☐ **Seminar**	
Module Code	**AIE221**				
ECTS Credits	**5**				
SWL (hr/sem)	**125 H/Sem.**				
Module Level	2	**Semester of Delivery**	4		
Administering Department	Artificial Intelligence Eng.	**College**	College of Engineering		
Module Leader	Firas Sabar Miftan	**e-mail**	firas@utq.edu.iq		
Module Leader's Acad. Title	Prof.	**Module Leader's Qualification**	Ph.D.		
Module Tutor	Ahmed Jabar	**e-mail**			
Peer Reviewer Name	Hasnaa Mohammad	**e-mail**			
Scientific Committee Approval Date		**Version Number**	1.0		

Relation with other Modules			
Prerequisite module	**AIE211**	**Semester**	3
Co-requisites module	None	**Semester**	

Module Aims, Learning Outcomes and Indicative Contents	
Module Objectives	**Recursion, Calculating the Sum of a List, Palindrome, Searching, Sequential Search, Binary Search, Hashing, Collision Resolution, Sorting (Bubble Sort, Selection Sort, Insertion Sort, Shell Sort, Merge Sort and Quick Sort) , Graphs, Trees, Binary Tree, Binary Search Tree, Tree Operations (Create, Insert, Delete)**

وزارة التعليم العالي والبحث العلمي
جامعة العين العراقية
كلية الهندسة
قسم الذكاء الاصطناعي

أ.م.د. سرمد كاظم دياب

Module Learning Outcomes	Knowledge	A1	**A1.1** Define the fundamental concepts of data structures and their role in efficient data storage and manipulation. **A1.2** Recognize various data structures such as arrays, matrices, stacks, queues, linked lists, trees, and graphs. **A1.3** Describe recursion techniques and their applications in problem solving, including Palindrome detection and list processing. **A1.4** Explain the principles and applications of Searching, including Sequential Search and Binary Search. **A1.5** Discuss Hashing techniques and Collision Resolution methods for efficient data retrieval. **A1.6** Choose appropriate Sorting algorithms (Bubble Sort, Selection Sort, Insertion Sort, Shell Sort, Merge Sort, Quick Sort) for different computational problems. **A1.7** Illustrate the use of Trees, Binary Trees, and Binary Search Trees for hierarchical data representation. **A1.8** Explain Tree Operations including creation, insertion, deletion, and traversal techniques.
		A4	A4.1 Identify the key concepts of data structures and understand their importance in programming tasks. A4.2 Apply recursion techniques in practical problems such as Palindrome checking and numerical computations. A4.3 Implement Searching algorithms (Sequential and Binary Search) in Python to solve real-world problems. A4.4 Demonstrate Hashing and Collision Resolution methods for managing large datasets. A4.5 Implement fundamental Sorting algorithms in Python and compare their efficiency and performance. A4.6 Construct and manipulate Trees, Binary Trees, and Binary Search Trees for hierarchical data. A4.7 Perform Tree Operations programmatically and analyze their computational complexity.

	Skills	B1	B.1.1 Function efficiently as an individual B.1.2 Function as a member of multidisciplinary and multicultural teams.
		B3	B.3.1 Acquire and apply new knowledge. B.3.2 Practice the strategies of searching for information in life – long self-learning
	Ethics	C3	C.3.1 Utilize contemporary technologies C.3.2 Make use of codes of practice and standards, quality guidelines.

Indicative Contents	Indicative content includes the following. **Unit 1:** Recursion Techniques, Calculating the Sum of a List, and Palindrome [4 hours] **Unit 2:** Searching Algorithms including Sequential Search, Binary Search, Hashing and Collision Resolution [6 hours] **Unit 3:** Sorting Algorithms including Bubble Sort, Selection Sort, Insertion Sort, Shell Sort, Merge Sort and Quick Sort [8 hours] **Unit 4:** Graphs, Trees, Binary Tree and Binary Search Tree [4 hours] **Unit 5:** Tree Operations (Create, Insert, Delete) and Applications of Tree Structures [4 hours]

Learning and Teaching Strategies

Strategies	• Presentation • Interactive learning • Discussions • Brainstorming • Group Solving Problems • Problem-Based Learning • Self-learning

Student Workload (SWL) for 15 weeks

Structured SWL (h/sem)	78	**Structured SWL (h/w)**	5.2

Unstructured SWL (h/sem)	47	**Unstructured SWL (h/w)**	4.8
Total SWL (h/sem)	125		

Module Evaluation تقييم المادة الدراسية					
		Time/Number	**Weight (Marks)**	**Week Due**	**Relevant Learning Outcome**
Formative assessment	**Quizzes**	2	10% (10)	5 and 10	A1, A4
	Assignments	2	10% (10)	2, 4,6,8,10,12 and 14	A1, A4
	Projects / Lab.				A1, A4
	Report	1	10% (10)	13	A4, B1, B3
Summative assessment	**Midterm Exam**	1hr	10% (10)	7	A1
	Final Exam	3hr	50% (50)	15	A1, A4
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus) المنهاج الاسبوعي النظري		
Week	**Material Covered**	**Los**
Week 1	Introduction to Recursion	A1.1, A4.1
Week 2	Calculating the Sum of a List and Palindrome	A1.1, A4.2, B1.1, B3.1, C3.1
Week 3	Introduction to Searching and Sequential Search	A1.2, A4.3, B1.1, B3.1, C3.1
Week 4	Binary Search	A1.2, A4.3, B1.1, B3.1, C3.1
Week 5	Hashing Concepts	A1.3, A4.4, B1.1, B3.1, C3.1
Week 6	Collision Resolution Techniques	A1.3, A4.4, B1.1, B3.1, C3.1
Week 7	Sorting Algorithms: Bubble Sort and Selection Sort	A1.4, A4.5, B1.1, B3.1, C3.1

Week 8	Sorting Algorithms: Insertion Sort and Shell Sort	A1.4, A4.5, B1.1, B3.1, C3.1
Week 9	Mid-term EXAM	A1, A4
Week 10	Sorting Algorithms: Merge Sort and Quick Sort	A1.4, A4.5, B1.1, B3.1, C3.1
Week 11	Graphs and Trees	A1.5, A4.6, B1.1, B3.1, C3.1
Week 12	Binary Tree and Binary Search Tree	A1.5, A4.6, B1.1, B3.1, C3.1
Week 13	Tree Operations (Create, Insert, Delete) and Final Review	A1.6, A4.7, B1.1, B3.1, C3.1
Week 14	Rubric Exam Preparatory week before the final Exam	A1, A4, B3

Delivery Plan (Weekly Lab. Syllabus)

المنهاج الاسبوعي للمختبر

Weeks	Material Covered	Los
Week 1	Recursion basics – calculating the sum of a list using recursion	A4.2, B1.1, B3.1, B3.2, C3.1
Week 2	Recursion applications – Palindrome checking	A4.2, B1.1, B3.1, B3.2, C3.1
Week 3	Searching algorithms – Sequential Search implementation	A4.3, B1.1, B3.1, B3.2, C3.1
Week 4	Searching algorithms – Binary Search implementation	A4.3, B1.1, B3.1, B3.2, C3.1
Week 5	Hashing concepts and simple hash function implementation	A4.4, B1.1, B3.1, B3.2, C3.1
Week 6	Sorting algorithms – Bubble Sort, Selection Sort, Insertion Sort implementation	A4.5, B1.1, B3.1, B3.2, C3.1
Week 7	Sorting algorithms – Shell Sort, Merge Sort, Quick Sort implementation	A4.5, B1.1, B3.1, B3.2, C3.1
Week 8	Mid-Term Exam	A1, B1
Week 9	Graphs and Trees – basic implementation and traversal concepts	A4.6, B1.1, B3.1, B3.2, C3.1
Week 10	Binary Tree and Binary Search Tree implementation	A4.6, B1.1, B3.1, B3.2, C3.1

Week 11	Tree Operations – Create, Insert, Delete	A4.7, B1.1, B3.1, B3.2, C3.1
Week 12	Mini Project / Practical Applications	A1, A4, B1, B3, C1
Week 13	Performance Evaluation and Final Review	A1, A4, B1, B3, C1
Week 14	Rubric Exam / Preparatory Week	A1, A4, B1

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	**Available in the Library?**
Required Texts	Miller, B. & Ranum, D. (2017). *Problem Solving with Algorithms and Data Structures*, Release 3.0.	NO
Recommended Texts	Karumanchi, N. (2017). *Data Structures and Algorithms Made Easy* (5th Edition).	NO
Websites	https://eng.alayen.edu.iq//colleges/e/	

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.