COURSE DESCRIPTIONS

Faculty	**Information Technology**				
Department	**Computer science**			**NQF level**	7
Course Title	**Data structures**	**Code**	**501241**	**Prerequisite**	501221
Credit Hours	**3hours**	**Theory**	2 h	**Practical**	1 h
Course Leader	Dr. Sami Qawasmeh	**email**	sqawasmeh@jadara.edu.jo		
Lecturers	**Dr. Ayat Al-jarrah**	**email**	ay.aljarrah@jadara.edu.jo		
	Dr. Sami Qawasmeh		sqawasmeh@jadara.edu.jo		
	Dr. Omer Rawashdeh		o.rawashdeh@jadara.edu.jo		
	Dr. Bushra Ghazala		bu.ghazala@jadara.edu.jo		
	Dr. Mohammad Albawaneh		m.Albawaneh@jadara.edu.jo		
	Dr. Tamer Baniamer		t.baniamer@jadara.edu.jo		
	Dr. Mashhour Beneaiesh		m.beneaiesh@jadara.edu.jo		
	Dr. Hadeel Alomari		h.alomari2@jadara.edu.jo		
Lecture time	Multiple Sections(12)	**Classroom**	Multiple Sections	**Attendance**	Fulltime
Semester	**First 2026-2027**	**Production**	20/6/2020	**Updated**	01/10/2026
Type of Teaching	☑ Face to Face ☐ Blended ☑ Online				

Short Description

This course is to introduce students to several types of data structures, their logical and physical representations, and their related operations. Topics to be covered include data structure, operations, lists and matrix representations, linked lists and their different variations, string storage representation and manipulation, queues and stacks implementation and application, Hash Structures: Representation, Search and Implementation and other issues, tree structures and their different variations.

Course Objectives

The main objective of this course is to provide students with a theoretical background and practical experience relating to the design and implementation of several types of data structures. Students, during this course, are trained to deal with Abstract Data Types (ADT) and to implement different data structures using different methods. Also, writing advanced programs which are based on arrays, structures, strings, lists, stacks, queues, hash tables, trees and recursion. In addition, apply appropriate data structures in problem solving.

Intended Learning Outcomes (CILOs)

A. Knowledge - Theoretical Understanding

a1. Define the basic concepts of data structures and their implementations and the importance of data structures in context of writing efficient programs with respect to time complexity (K1).

B. Knowledge - Practical Application
a2. Demonstrate the use of pointers and dynamic memory management to implement and manipulate dynamic data structures, including linked lists, stacks, queues, and trees. (K2).
C. Skills - Generic Problem Solving and Analytical Skills
b1. Develop applications based on using different data structures to determine the appropriate data structure(s) based on the advantages and disadvantages of specific algorithms (S1).
D. Skills - Communication, ICT, and Numeracy
E. Competence: Autonomy, Responsibility, and Context
c1. Apply appropriate data structures in problem solving for the best solution. (C1)
Teaching and Learning Methods
☑ Face to Face Lectures ☑ Brainstorming Synchronous remote ☐ Asynchronous remote. ☐ Using Video ☑ Discussions ☐ Research Project ☐ Case Study ☐ Field visit ☑ Problem solving
Assessment Methods
☑ Formative Assessment ☑Quiz ☑ Lab Exam ☑ Homework ☐ Project Assessment ☐ Oral Presentation ☑ Midterm ☑ Final Exam

Course Contents					
Week	**Hours**	**CILOs**	**Topics**	**Teaching & Learning Methods**	**Assessment Methods**
1	4	a1	• *Introduction: Data Structures* • *Classes and Objects:* - *Create a Class* - *Create an Object* - *Array of objects* - *Size of an Object*	Cooperative Learning, Program Examples	
2	4	a1	• *Pointers Operators. Pointer Expressions and pointer arithmetic, Relation between pointers and Arrays*.	Cooperative Learning, Program Examples	
3	4	a1	• *Structures –Declaration, initialization, accessing structures, operations on structures, Complex structures, structures and functions, Passing structures through pointers.*	Cooperative Learning, Program Examples	homework
4	4	a1, a2	• *Dynamic structures: Pointer Variable Definitions and Initialization, Dynamic memory allocation - Pointers and classes.*	Cooperative Learning, Program Examples	Quiz
5-8	16	a:1,2 b1, c1	• *Linked List:* Definition, Basic terminology, *Implementation of Linked List, Linked List Vs. Arrays, Memory Allocation & De-allocation*	Inquiry-Based Learning,	Homework Midterm Assignment

			for a Linked List, Linked List operations (insertion, deletion), Types of Linked List, and Application of Linked lists. • Introduction to complexity: Big-O notation • Big O()-Notation	Program Examples	
9-11	12	a:1,2 b1 c1	• *Linked list (double, Bidirectional, Circular) Linked-list, operations, and Applications on linked-list, sorted and unsorted linked-lists*	Blended Learning, Program Examples	Homework Assignment
12-14	12	a:1,2 b1 c1	• Stacks and Queues: Definition, Basic terminology, Behavior of a Stack. Basic operations on a Stack. Linked-list based implementation. Expression evaluation using a stack. Parentheses matching. • Queues. Behavior of a queue. Basic queue operations Study implementations using a linked list. *Queues- (Priority Queues), Double-Ended Queues, Circular Queues*	lectures, Program Examples	Quiz and assignment
15	4	a:1,2 b1 c1	• Binary Trees: Definitions for Binary Tree and Binary Search Tree Implementing Binary Trees using Linked Nodes Implementing a List Properties of Binary Trees Full, Balanced, Complete Binary Trees	lectures, Program Examples	Quiz and assignment
16	4	c1	• Graphs: Graph representations and Graph traversal • **BFS** method **DFS** method	Inquiry-Based Learning	discussion

Infrastructure	
Textbook	**D. S. Malik, "C++ Programming: Program Design Including Data Structures, 8th edition", 2018.**
References	**C++ How to program, H. M. Deitel and P. J. Deitel, Pearson India Education; 10th ed, 2017.**
Required reading	
Electronic materials	**Posted materials**
Other	**Handout materials**

Assessment Method		Grade	a1	a2	b1	c1
First (Midterm)		30	9	8	7	6
Second (if applicable)		-	-	-	-	-
Final Exam		40	9	9	11	11
Coursework						
Coursework assessment methods	Assignments	10	2.5	2.5	2.5	2.5
	Case study					
	Discussion and interaction					
	Group work activities					
	Lab tests and assignments	10	2.5	2.5	2.5	2.5
	Presentations					
	Quizzes	10	2.5	2.5	2.5	2.5
Total		100	28	28	23	21

Plagiarism
Plagiarism is claiming that someone else's work is your own. The department has a strict policy regarding plagiarism and, if plagiarism is indeed discovered, this policy will be applied. Note that punishments apply also to anyone assisting another to commit plagiarism (for example by knowingly allowing someone to copy your code). Plagiarism is different from group work in which many individuals share ideas on how to do the coursework. You are strongly encouraged to work in small groups, and you will certainly not be penalized for doing so. This means that you may work together on the program. It is important that you fully understand all aspects of the completed program. In order to allow proper assessment that this is indeed the case, you must adhere strictly to the coursework requirements as outlined above and detailed in the coursework problem description. These requirements are in place to encourage individual understanding, facilitate individual assessment, and deter plagiarism.