Identification	**Subject**	CMS 215 Data Structures 3 credits, 3KU /6ECTS credits
	Department	Computer Science
	Program	Undergraduate
	Term	Fall, 2023
	Instructor	Mohammad Ali AL-Qudah
	E-mail:	mohammad.ali@khazar.org
	Phone:	(+994 51- 3053089)
	Classroom/hours	41 Mehseti str. (Neftchilar campus), Thursday 17:00-18:30, Monday 15.20:00-18:30
Prerequisites	CMS 205 Object Oriented Programming	
Language	English	
Compulsory/Elective	Required	
Required textbooks and course materials	***Core Textbook:*** 1. Data Structures Using C++, Second Edition D.S. Malik **Supplementary Textbooks** *2. Data Structures and Algorithms in C++, 2nd edition by Michael T. Goodrich, Roberto Tamassia, David M. Mount*	
Course website	This course combines traditional face-to-face classes.	
Course outline	This course introduces the students to data structures using an object-oriented programming language. This includes logical and physical representation of data structures, collection types, array-based lists, linked lists, stacks, queues, and basics of algorithm analysis, binary trees, binary search trees, hashing, and heaps. Applications and algorithms based on data structures are covered in this course. Throughout the semester, problem-solving skills will be stressed and applied to solving computing problems. Weekly homework experiments will provide hands-on experience in topics covered in this course.	
Course objectives	By the end of this course, students should be able to: • Understand algorithms, time complexity and space calculating. • Understand the sorting and searching fundamentals. • Describe and /or define the Abstract Data Types, including lists, stacks, queues, trees, hash tables and graph. • Understand, explain, demonstrate, and evaluate alternate implementations of examples of the methods associated with Abstract Data Types. • Implement and test Abstract Data Types in generic programs using C++.	

Learning outcomes	By successfully completing this course, students will be able to demonstrate the following outcomes: 1. Explain the concept of “Data Structure”, “Abstraction.” 1.1 Understanding the difference between Data Structure and Abstraction 1.2 Understand the difference between Data structure and programming language. 2. Explain the “Abstract Data Type” concept, the different ADT types, and their implementation. 2.1 Understanding the concept of an Abstract Data Type (ADT). 2.2 Describe the properties, interfaces, and behaviors of basic abstract data types, such as List, Sorted List, Stack, Queue, and Tree 3. Enabling students to design and implement some user-defined data structures for the different ADT as C++ classes. 3.1 Compare and contrast the operation of common data structures (such as linear structures, priority queues, and tree structures, in terms of abstract data type operations and their implementation 3.2 Have a good knowledge in which data structure should be used with each ADT. 4. Giving practice in the application of new user-defined data structures. 4.1 Solve problems computationally through the application of fundamental data structures. 4.2 Improving the programming skills of students in the use of some object-oriented programming language (C++).

Teaching methods	**Lecture**		X
	Group discussion		X
	Experiential exercise		X
Evaluation	**Methods**	**Date/deadlines**	**Percentage (%)**
	Midterm Exam		30
	Project		20
	Final Exam		40
	Homework		10
	Total		100
Policy	**Project description** In the projects, there are applications in practical life. Students will build these applications in several stages. It is possible to benefit from these ideas or any new idea shown on the data structure we use C++.Students should present their topic by the end of December. • A hash table (uniquely identifies each feed while allowing additional feeds to be added (assuming dynamic resizing)) • Linked List (doubly linked: from one node, you can go backward/forwards one by one) • Graph (each person is a point, and connections/friendships are an edge) • Array (2-dimensional, 1000×1000, storing color values) • Queue (like a queue/line of people waiting to get through a checkpoint)		

- **Preparation for class**

The structure of this course emphasizes the importance of independent study and preparation outside of class. The lecture material will concentrate on the key points raised in the text. Reading the assigned chapters and becoming acquainted with them prior to class will aid your understanding of the lecture. Following the lecture, you should review your notes and work on relevant problems and cases from the chapter's end, as well as sample exam questions.

We will also have many review sessions throughout the semester. These review sessions will take place during the regular class times.

- **Withdrawal (pass/fail)**

This course strictly adheres to the grading policy of the School of Engineering and Applied Science. As a result, a student is normally expected to pass with a grade of at least 60%. In the event of failure, he or she will be required to repeat the course the following term or year.

- **Cheating/plagiarism**

Cheating or other plagiarism during the Quizzes, Mid-term and Final Examinations will lead to paper cancellation. In this case, the student will receive a zero (0) without further consideration.

- **Professional behavior guidelines**

During class, students must act in a way that fosters a positive academic and professional environment. Unauthorized conversations and unethical behavior are forbidden.

- **Ethics**

Students should not arrive in late to class.

All cell phones must be turned off and stowed away before entering class.

Use of any electronic devices is not allowed in the classroom and violators will be punished accordingly.

Tentative Schedule

Week	**Date/Day (tentative)**	**Topics**	**Textbook**
1		Overview of Data Structure and Algorithms • What is data structure? • What to learn from this course (Project Description	Ch. 1 From Textbook
2		• Array and Linked List • Elementary Data Structures: • Stacks and queues • Linked lists • Implementing pointers and objects • Representing rooted trees	Ch. 3 From Textbook
3		Big O Notation • Asymptotic notation • Standard notations and common functions 1st quiz and First Assignment	Ch. 1 From Textbook

4		Stacks • Stack In C++ • Basic Operations • Illustration • Implementation • Using Arrays • Using A Linked List • Applications of Stack • Infix To Postfix Expressions • Expression Parsing/Evaluation • Tree Traversals • Sorting Algorithms • Towers Of Hanoi	Ch. 7 from Textbook
5		Start Queues •What is a queue in C++? •How to use a C++ queue •FIFO Queues •Priority Queues •When to use queues in C++ •When to avoid using queues in C++ •Underlying containers in C++	Ch. 8 From Textbook
6		Continue Queues & List and Sequence	Ch.8 From Textbook
7		Recursion • What Is Recursion in C++? • Why Do We Need Recursion? • Working of Recursion • Types of Recursions	Ch 6 From Textbook
8		Searching algorithms and Hashing • Direct-address tables • Hash tables • Hash functions • Open addressing	Ch 9 From Textbook
9		Review for Midterm exam & Exercise. Midterm Exam.	
10		Continue>> Searching algorithms and Hashing	Ch 9 From Textbook
11		Binary Trees • What is a binary search tree? • Querying a binary search tree • Insertion and deletion • Properties of red-black trees • Rotations • Insertion • Deletion	Ch11 From Textbook
12		Continue>> Binary Trees	Ch 11From Textbook

13		Graphs • Breadth First Traversal for a Graph • Depth First Traversal for a Graph • Applications of Breadth First Traversal • Iterative Depth First Search	Ch 12 From Textbook
14		Sorting Algorithms • Lower bounds for sorting • Counting sort • Radix sort • Bucket sort	Ch10 From Textbook
15		Continue Sorting Algorithms Final Exam Review	Ch10 From Textbook
		Final Exam	

syllabus is a guide for the course and any modifications to it will be announced in advanc