Identification	**Department**	Computer Science
	Program	Undergraduate
	Subject	CMS 215 Data Structures 3 credits
	Term	Spring 2023
	Instructor	Mohammad AL-Qudah
	E-mail	Mohammad.ali@khazar.org
	Classroom/hours	11 Mehseti str. (Neftchilar campus)
Prerequisites	CMS 205 Object Oriented Programming	
Language	English	
Compulsory/Elective	Compulsory	
Text books and course materials	**Core Textbooks**: **Data Structures Using C++, Second Edition D.S. Malik** **Supplementary Textbooks**: 1. *Data Structures and Algorithms in C++, 2nd edition by Michael T. Goodrich, Roberto Tamassia, David M. Mount* 2. Introduction to Algorithms, 3rd Edition, (2009) by Thomas H. Cormen, Charles E. Leiserson, 3. C++ programming : From Problem Analysis to Program Design /C plus plus programming. : Malik, D S. Course Technology, Boston, MA 4. Problem solving with C++ / Savitch, Walter J, 1943- Pearson/Addison-Wesley, Boston : TENTH EDITION Edition. 5. Class notes 6.https://www.w3schools.com/ 7.https://www.tutorialspoint.com/	
Teaching methods	**Case analysis**	
	Group discussion	**+**
	Lab	**+**
	Lecture	**+**
	Course paper	**+**

	Others	
Evaluation Criteria	**Methods**	**Percentage (%)**
	Midterm Exam	**30%**
	Case studies	
	Activity	**5%**
	Quizzes (4)	**15%**
	Project (1)	**10%**
	Presentation	
	Laboratory Work (Assignments)	
	Final Exam	**40%**
	Other	
	Total	**100%**
Course objectives	1. Understand algorithms, time complexity and space calculating 2. Understand the sorting and searching fundamentals. 3. Describe and /or define the Abstract Data Types; including lists, stacks, queues, trees, hash tables and graph. 4. Understand, explain, demonstrate, and evaluate alternate implementations of examples of the methods associated with Abstract Data Types. 5. Implement and test Abstract Data Types in generic programs using C++.	
Learning outcomes	By successfully completing this course, students will beable to: 1. Ability to analyze algorithms and algorithm correctness. 2. Ability to summarize searching and sorting techniques 3. Ability to describe stack, queue, and linked list operations. 4. Ability to have knowledge of tree and graphs concepts.	

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 policy	***Attendance*** *Attendance is very important for the course. In accordance with university policy, students missing more than 25% of total classes are subject to failure. Penalties may be assessed without regard to the student's performance. Attendance will be recorded at the beginning or end of each class.* ***Exams*** *All exams will be CLOSE-BOOK; necessary algorithms/equations/relations will be supplied as convenient.* ***Make ups:*** *Unless arrangements are worked out in advance, missed assignments **cannot** be made up, and 10% per week will be deducted for late submissions. Exams' makeup must go through the department and faculty approvals process.* ***Homework and Assignments:*** *Several Lab assignments will be assigned. Most assignments will be submitted electronically using the teams or email system. **Homework and assignments must be submitted before their corresponding due date and time, as indicated in the assignment description.** Late submissions will be assessed a 10% penalty per day. All Lab Assignments must be submitted before the start of the next lab week and will not be graded thereafter.*

	In-Class Computers and Handheld Devices: *Phone calls, text messages, instant messages, email, and general web surfing are **not allowed** during class time. Computers may only be used to follow the material in class. Violators will have their devices confiscated or asked to leave the room* ***Cheating:*** *In this course, **all** assignments, exams, and project submissions implicitly imply that it is the sole work of the author, unless joint work is explicitly authorized. Help may be obtained from the instructor or other students to understand the description of the problem and any technology, but the solution must be the student's own work. If joint work is authorized, all contributing students must be listed on the submission. Any deviation from this is considered a cheating attempt, and as a minimum, will result in failure of the submission and as a maximum reporting the incident to the department and the faculty to apply the University rules.*

Tentative Schedule			
Week	Date/Day (tentative)	Topics	Textbook Assignments
1		**Overview of Data Structure and Algorithms** • What is data structure? • What to learn from this course (Theory: 90 minutes + Practice: 45minutes + Group discussion: 45 minutes)	Chapter 1 From Textbook
2		**Array and Linked List** • Elementary Data Structures: • Stacks and queues • Linked lists • Implementing pointers and objects • Representing rooted trees (Theory: 90 minutes + Practice: 90minutes)	Chapter 3 From Textbook
3		**Big O Notation** • Asymptotic notation • Standard notations and common functions **1st quiz and First Assignment** Theory: 90 minutes + Practice: 45 minutes + Quiz#01: 45 minutes)	Chapter 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 (Theory: 90 minutes + Practice: 90 minutes)	Chapter 7 From Textbook
5		**>>Continue Stacks** **Start Queues** • What is a queue in C++? • How to use a C++ queue • FIFO Queues	Chapter 8 From Textbook

		• Priority Queues • When to use queues in C++ • When to avoid using queues in C++ • Underlying containers in C++ **2nd Quiz** Theory: 90 minutes + Practice: 45 minutes + Quiz#02: 45 minutes)	
6		**>>Continue** Queues & List and Sequence **Second Assignment** (Theory: 90 minutes + Practice: 45 minutes + Group discussion: 45 minutes)	Chapter 8 From Textbook
7		**Recursion** • What Is Recursion in C++? • Why Do We Need Recursion? • Working of Recursion • Types of Recursion **Review midterm exam** (Theory: 90 minutes + Practice: 90minutes)	Chapter 6 From Textbook
8		**Midterm Exam**	
9		**Searching algorithms and Hashing** • Direct-address tables • Hash tables • Hash functions • Open addressing (Theory: 90 minutes + Practice: 90 minutes)	Chapter 9 From Textbook
10		**>>continue Searching algorithms and Hashing** **Third Assignment** (Theory: 90 minutes + Practice: 90minutes)	Chapter 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	Chapter 11 From Textbook

		• Rotations • Insertion • Deletion (Theory: 90 minutes + Practice: 45 minutes + Group discussion: 45 minutes)	
12		**>>Continue Binary Trees** **3rd quiz** (Theory: 90 minutes + Practice: 45minutes + Quiz#03: 45 minutes)	Chapter 11 From Textbook
13		**Graphs** • Breadth First Traversal for a Graph • Depth First Traversal for a Graph • Applications of Breadth First Traversal • Iterative Depth First Search (Theory: 90 minutes + Practice: 90minutes)	Chapter 12 From Textbook
14		**Sorting Algorithms** • Lower bounds for sorting • Counting sort • Radix sort • Bucket sort **4th Quiz** Theory: 90 minutes + Practice: 45minutes + Quiz#04: 45 minutes)	Chapter10 From Textbook
15		>>Continue Sorting Algorithms (Theory: 45 minutes + Practice: 45 minutes + Project Presentation: 75 minutes + Exam Preparation: 15 minutes)	Chapter 10 From Textbook
16		**Final Exam**	

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