CMPE/CMSE 231 Data Structures

Department: Computer Engineering

Instructor Information
Name: Prof.Dr. Marifi Güler
Office: CMPE 224

Assistant Information
Name: Zeinab Jabbarbabouei (Coord)
Office: CMPE102

Program Name: Computer Engineering	**Program Code:** 25/29

Course Code	**Credits**	**Year/Semester**
CMPE 231	4	2023-2024 Spring

☒ Required Course ☐ Elective Course

Prerequisite(s):
CMSE112 Introduction to Programming

Catalog Description

Introduction to Data Structures: Primitive data structures. Binary and Decimal Integers, Real numbers, Character strings, Memory representation of information.

Arrays and Memory allocation (storage) of arrays. Character string operations. Two and multi dimensional arrays.

Structures: Arrays of structures. Self-referential structures. Structures and Functions. Dynamic memory allocation .

The Stack : Stack as an Abstract Data Type. Primitive operations. Representing the stack in C. Infix, Postfix, and Prefix notations; Infix-to-Postfix conversion.

Recursion : Recursive definition. Examples: Factorial function. Fibonacci sequence. Binary search. The Towers of Hanoi problem. Recursion versus Iteration.

Queues : The Queue as an Abstract Data Type. C implementation of Queues.

Linked Lists: Inserting and Removing Nodes from a List. Linked implementation of Stacks and Queues. Linked Lists using Dynamic Variables. Queues as Lists in C. Circular Lists(Stack as a Circular List Queues as a Circular List)**,**Doubly Linked Lists.

Trees: Operations on Binary Trees. Binary Tree Representations. Binary Tree Traversals. Creating a binary tree.

Sorting : Efficiency of Sorting. The O notation. Bubble Sort. Quick Sort.

Searching : Sequential Search. Binary Search. Binary Search Trees.

Course Web Page
https://staff.emu.edu.tr/marifiguler/en/teaching/cmpe231

Textbook(s)

Langsam Y., Augenstein M., Tenenbaum A.
Data Structures Using C and C++, 2nd edition, Prentice Hall Int., 1996
(ISBN 013-529322-7)*From the Publisher (Prentice Hall/Engineering)*

Indicative Basic Reading List

Loudon, K. [http://www.oreilly.com/catalog/masteralgoc/]
Mastering Algorithms with C, 1st edition, O'Reilly & Assoc., 1999.

Esakov, J., Weiss, T. Data Structures: An Advanced Approach Using C, Prentice Hall Software Series, 1989.

Weiss, M.A. Data Structures and Algorithm Analysis in C, 2nd edition, Addison-Wesley Pub. Co., 1996.

Tremblay J.P., Sorenson, P.G. An Introduction to Data Structures with Applications, 2nd edition, McGraw-Hill Inc, 1984.

Kruse, R.L. Data Structures and Program Design, 3rd edition, Prentice Hall Inc., 1994.

Lipschutz, S. Theory and Problems of Data Structures, McGraw-Hill, Inc., 1986.

Topics Covered and Class Schedule(4 Hours of classroom teaching each week)

- **weeks 1-2 Primitive data structures**. Binary and Decimal Integers, Real numbers, Character strings, Memory representation of information, pointers.
- **week 3-4 Arrays** and Memory allocation (storage) of arrays. Character string operations. Two and multi dimensional arrays. **Structures** (Arrays of structures. Self-referential structures. Structures and Functions). Dynamic memory allocation. First Assignment is given for duration of two weeks.
- **weeks 5-6 The Stack :**Stack as an Abstract Data Type. Primitive operations. Representing the stack in C. Infix, Postfix, and Prefix notations; Infix-to-Postfix conversion..
- **weeks 7-8 The stack and Recursion :** Recursive definition. Examples: Factorial function. Fibonacci sequence. Binary search. The Towers of Hanoi problem. Recursion versus Iteration (1.Midterm Exam)
- **weeks 9 Queues :**The Queue as an Abstract Data Type. C implementation of Queues. Circular queu representation.
- **weeks 10-12 Linked Lists**: Representation of linked list structures. Main operations using linked list structures. Type of linked list structures. Representation of Stacks and Queues using linked list. Linked Lists using Dynamic Variables. Queues as Lists in C. Circular Lists(Stack as a Circular List Queues as a Circular List)**,** Doubly Linked Lists. 2.Assigment is given for duration of 2 weeks.(2.Midterm examination)
- **weeks 13-14 Tree representation,** Binary Tree Representations. Operations on Binary Trees. Binary Tree Traversals. Creating a binary tree. Deleting nodes from a binary tree. Sorting and Searching

week 15 Review and Discussion

Lab Schedule

Week of	
2024 March 4	Experiment 1
2024 March 11	Experiment 2
2024 April 15	Experiment 3
2024 May 6	Experiment 4
2024 May 13	Experiment 5
2024 May 20	Experiment 6

Course Learning Outcomes

On successful completion of the course, the student is expected to be able to:

- Use the C programming language in the implementation of data structures
- Develop recursive algorithms and functions
- Implement and use the stack abstract data type
- Implement and use the queue abstract data type
- Implement and/or use the linear linked list abstract data type
- Implement and/or use the circular linked list abstract data type

- Implement and/or use binary search trees
- Implement and/or trace the execution of sort algorithms
- Implement and/or trace the execution of the binary search algorithm
- Perform ordered traversals of trees (in-order, pre-order, post-order)
- Perform infix-postfix conversion of arithmetic expressions

	Method	**No**	**Percentage**
Assessment	Midterm Exam(s)	1	35%
	Final Examination	1	50%
	Lab work	6	15%

Student should check course web site for the laboratory experiments and assignments at least one week in advance of the relevant laboratory work.

Only one makeup exam will be given for the midterm exams or the final exam in total.

If a student misses any examination, **S/He MUST submit a written medical report** to the course coordinator stating their excuse, within 3 days of that examination in order to be eligible for the makeup exam.

NG grade will only be given if a student does not have any exam grade.

The students who repeat this course are not exempted from the lab work**s.**
There will be NO make-up from lab. The sum of the highest 5 grades will be taken as the overall lab grade.

Students are expected to attend the lectures although no points will be given for the attendance.

Relationship of the course to ABET Student Outcomes

1. an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics

7. an ability to acquire and apply new knowledge as needed, using appropriate learning strategies.

Prepared by: Prof. Dr. Marifi Güler	**Date Prepared:** 28.2.2024