Eastern Mediterranean University
Faculty of Engineering, Department of Electrical and Electronic Engineering
EENG212/INFE212 – Algorithms and Data Structures
Web: http://opencourses.emu.edu.tr/course/view.php?id=286

COURSE DESCRIPTION

Year and Semester : 2, FALL, SPRING
Credit Hour : (4, 1) 4
Pre-requisite(s) : EENG112/ INFE112
Academic Term : Spring 2016-2017

Catalog Description:

Storage structures and memory allocations. **Primitive data structures**. Data abstraction and Abstract Data Types. **Array and record structures**. Sorting algorithms and quick sort. **Linear & binary search**. Complexity of algorithms. **String processing**. Stacks & queues; stack operations, implementation of recursion, polish notation and arithmetic expressions. **Queues and implementation methods**. Dequeues & priority queues. **Linked storage representation and linked-lists**. Doubly linked lists and circular lists. **Binary trees**. Tree traversal algorithms. **Tree searching. General trees**. Graphs; terminology, operations on graphs and traversing algorithms.

Prerequisite by Topic:

Detailed knowledge and experience on C-Language as a high-level programming language.

Instructor:

Prof. Dr. Hasan AMCA Office Tel: 630 1500 / 630 3200 e-mail: hasan.amca@emu.edu.tr	**Office Hours**	
	Tue.	08.30 – 10.20
	Thu.	08.30 – 10.20

Lab Assistant(s): Please see the lab web page for details at **http://faraday.ee.emu.edu.tr/ee212-lab** or follow the ling from the course web page.

Textbook(s):

For this course there will be NO main textbook and the notes prepared by the lecturer will be used. However, most of the concepts studied can be found in the book "*C How to Program, H. M. Deitel and P. J. Deitel, Prentice Hall, 5th (or above) edition*".

References:

1. Data Structures using C and C++, Y. Langsam , M. Augenstein, A. Tenenbaum, 2nd ed. Prentice Hall, 1996

Course Objectives:

At the end of this course, students will be able to:

1. *understand* the fundamental data structures and Abstract Data Types,
2. *understand* the main sorting and searching algorithms and recursion,
3. *analyze* the time and space complexity of a given algorithm,
4. *understand and implement* stacks and queues,
5. *process* the linked list and tree structures,
6. *understand the graph terminology and perform basic graph operations.*

COURSE OUTLINE & ORGANIZATION

Review of fundamental topics in C: Functions; call by value and call by reference, Arrays; single subscripted arrays and multiple subscripted arrays, Pointers, Structures. Dynamic memory allocation.
Structures Defining structures, accessing members using dot or arrow operators, typdef usage, examples.
The Stack and Recursion Stack as an Abstract Data Type, primitive Stack operations, Representing Stack in C, Examples
MIDTERM Exam I
Linked Lists Insertion and deletion, Linked implementation of Stacks and Queues, Linked List as a Data Structure, array implementation of Lists. Other List Structures; Circular Lists, Doubly Linked Lists.
Queues The Queue as an Abstract Data Type, C implementation of Queues, Priority Queue
Trees Binary Trees, Binary Tree representation, Representing Lists as Binary Trees, Trees and their Applications; Tree Searching, Tree Traversals, insertion and deletion.
FINAL Exam

Laboratory Work: Laboratory sessions include C-Language as a high level programs for the implementation of data structures. One assignment is given for each fundamental type.

Grading Policy

Homework	: 5%,
QUIZ	: 10%,
Midterm I	: 30%,
Laboratory	: 15%,
Final Examination	: 40%.

Exemption Policy:

Students must contact their lecturer for exemption during first week of the semester.

NG Policy:

1. **Students who do not attend more than 60 % of the course lecture hours will be given NG grade.**
2. **Students who miss more than 2 weeks of lab sessions will also be given NG grade.**

Makeup and Re-sit Policies

Students with an NG grade cannot take the make-up or re-sit exam.

Relationship of Course to Program Outcomes

The course has been designed to contribute to the following program outcomes:

1. An ability to apply knowledge of mathematics, science, and engineering
2. An ability to design a system, component , or process to meet desired needs within realistic constraints
3. An ability to identify , formulate and solve engineering problems
4. An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.