EENG212 : Algorithms & Data Structures

Department:
Electrical and Electronic Engineering

Program Name:
Electrical and Electronic Engineering, Information System Engineering

Program Code: 27

Course Number: EENG212 | **Credits:** 4

☒ Required Course ☐ Elective Course (click and check the appropriate box)

Prerequisite(s): EENG112

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.

Course Web Page:
Opencourses.emu.edu.tr → courses → EENG212

Textbook(s):
1) 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".

Indicative Basic Reading List :

1) Data Structures using C and C++, Y. Langsam , M. Augenstein, A. Tenenbaum, 2nd ed. Prentice Hall, 1996
2) Theory and Problems of Data Structures, S. Lipschutz, international edition, McGraw Hill, 1986.
3) Data Structures and Algorithm Analysis in C++, M. A. Weiss,Addison-Wesley, 1994.

Course Organization:

Week 1 3 Oct. 22	**Review of Teaching Environment and Techniques** **Review of Functions in C:** Functions; call by value and call by reference
Week 2 10 Oct. 22	**Review of Functions in C:** Functions; call by value and call by reference **Review of Arrays in C:** Arrays; single subscripted arrays and multiple subscripted arrays.
Week 3 17 Oct. 22	**Review of Arrays in C:** Arrays; single subscripted arrays and multiple subscripted arrays. **Review of Pointers in C:** Pointers, Structures, Dynamic memory allocation.
Week 4 24 Oct. 22	**Review of Pointers in C:** Pointers, Structures, Dynamic memory allocation. **Tutorial on Functions, Arrays and Pointers**
Week 5 31 Oct.22	**Introduction to Data Structures in C** **Introduction to Data Structures in C**
Week 6 7 Nov. 22	**Basic Data Architectures and Data Abstraction:** Integers, floating point numbers, characters, strings, Abstract Data Types. **Linked Lists in C:** Linked Lists, insertion and deletion, Linked implementation of Stacks, Linked List as a Data Structure, array implementation of Lists. Other List Structures; Circular Lists, Doubly Linked Lists
Week 7 14 Nov. 22	**Linked Lists in C:** Linked Lists, insertion and deletion, Linked implementation of Stacks, Linked List as a Data Structure, array implementation of Lists. Other List Structures; Circular Lists, Doubly Linked Lists.
Week 8 21 Nov. 22	**Mid-Term Exam Week** **21 Nov. 22 – 3 Dec. 22**
Week 9 5 Dec. 22	**The Stack and Recursion:** Stack as an Abstract Data Type, primitive Stack operations, Representing Stack in C, definition of recursion, recursive functions, and recursion versus iteration **Queues in C:** The Queue as an Abstract Data Type, C implementation of Queues, Priority Queue
Week 10 12 Dec. 22	**Trees:** Binary Trees, Binary Tree representation, Representing Lists as Binary Trees, Trees and their Applications; Tree Searching, Tree Traversals, insertion and deletion. **Trees…**

Week 11 19 Dec. 22	**Pseudo Code, Algorithms, Flowchart, Trace Table.** **Pseudo Code, Algorithms, Flowchart, Trace Table.**
Week 12 26 Dec. 22	**Algorithm Complexity** **Algorithm Complexity**
Week 13 2 Jan 23	**Tutorials Week**
Week 14 9 Jan 23	**Final Exam Week** **9-24 January**

Course Learning Outcomes:

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.*

Class Schedule:	**Laboratory Schedule:**
4 hrs of lectures per week	2 hrs of laboratory per week

	Method	**No**	**Percentage**
Assessment	Midterm Exam(s)	1	30%
	Lab Work(s)	7	15 %
	Quiz(s)	3-4	10 %
	Homework(s)	4	5%
	Final Examination	1	40%

Contribution of Course to Criterion 5

Credit Hours for:
Mathematics & Basic Science : 0
Engineering Design : 4
General Education : 0

Relationship of Course to Student Outcomes

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

(1) an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics
(2) an ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors
(6) an ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions.

Contribution of Course Learning Outcomes to Student Outcomes

	Student Outcome:						
Course Learning Outcome	**1**	**2**	**3**	**4**	**5**	**6**	**7**
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	•	•				•	

Updated by: Hasan Amca	**Date Prepared: 14.4.2021**