EE2331: DATA STRUCTURES AND ALGORITHMS

Effective Term
Semester B 2022/23

Part I Course Overview

Course Title
Data Structures and Algorithms

Subject Code
EE - Electrical Engineering
Course Number
2331

Academic Unit
Electrical Engineering (EE)

College/School
College of Engineering (EG)

Course Duration
One Semester

Credit Units
3

Level
B1, B2, B3, B4 - Bachelor's Degree

Medium of Instruction
English

Medium of Assessment
English

Prerequisites
CS2311 Computer Programming or equivalent

Precursors
Nil

Equivalent Courses
Nil

Exclusive Courses
Nil

Part II Course Details

Abstract

This aim of this course is to provide students with an understand of fundamental concepts of data structures and algorithm design, and to cultivate systematic programming discipline.

Course Intended Learning Outcomes (CILOs)

	CILOs	Weighting (if app.)	DEC-A1	DEC-A2	DEC-A3
1	apply structural programming approach to solve computation problems		x	x	x
2	demonstrate applications of standard data structures such as list, heap, tree, and graph		x	x	x
3	solve computation problems using recursion where appropriate		x	x	x
4	apply different sorting and searching algorithms		x	x	x

A1: Attitude
Develop an attitude of discovery/innovation/creativity, as demonstrated by students possessing a strong sense of curiosity, asking questions actively, challenging assumptions or engaging in inquiry together with teachers.

A2: Ability
Develop the ability/skill needed to discover/innovate/create, as demonstrated by students possessing critical thinking skills to assess ideas, acquiring research skills, synthesizing knowledge across disciplines or applying academic knowledge to real-life problems.

A3: Accomplishments
Demonstrate accomplishment of discovery/innovation/creativity through producing /constructing creative works/new artefacts, effective solutions to real-life problems or new processes.

Teaching and Learning Activities (TLAs)

	TLAs	Brief Description	CILO No.	Hours/week (if applicable)
1	Lecture	Explain key concepts in data structures and algorithm design.Explain implementation details in the C/C++ language.	1, 2, 3, 4	3 hrs/wk
2	Tutorials and assignments	Provide students with hands on and practical experiences in programming.Provide students with training in problem solving.	1, 2, 3, 4	1 hr/wk

Assessment Tasks / Activities (ATs)

	ATs	CILO No.	Weighting (%)	Remarks (e.g. Parameter for GenAI use)
1	Tests (min.: 2)	1, 2, 3, 4	25	
2	#Assignments (min.: 3)	1, 2, 3, 4	25	

Continuous Assessment (%)
50

Examination (%)
50

Examination Duration (Hours)
2

Additional Information for ATs
Remark:
To pass the course, students are required to achieve at least 30% in course work and 30% in the examination.

may include homework, tutorial exercise, project/mini-project, presentation

Assessment Rubrics (AR)

Assessment Task
Examination

Criterion
Achievements in CILOs

Excellent (A+, A, A-)
High

Good (B+, B, B-)
Significant

Fair (C+, C, C-)
Moderate

Marginal (D)
Basic

Failure (F)
Not even reaching marginal levels

Assessment Task
Coursework

Criterion
Achievements in CILOs

Excellent (A+, A, A-)
High

Good (B+, B, B-)
Significant

Fair (C+, C, C-)
Moderate

Marginal (D)

Basic

Failure (F)

Not even reaching marginal levels

Part III Other Information

Keyword Syllabus

Introduction

Overview of data types and data structures; Control structure, pointers in C/C++; Linear and multi-dimensional arrays; Parameter passing in function call; Review of structured programming; Introduce concepts of data encapsulation and program invariants; Class and object in C++.

Analysis of Algorithms

Overview of complexity analysis; Introduce the big-O notation; Asymptotic Complexity; Best, average and worst cases.

One dimensional data structure

Such as linked list/array/stacks/queues and their applications; Overview of the C++ STL.

Recursion

Introduce the concept of recursion; Examples of recursive algorithms: factorials, Ackerman function, recursive binary search, towers of Hanoi, etc; Recursion and backtracking.

Trees

Binary tree; Tree traversals; Example algorithms for tree operations; Applications: Huffman tree; Binary search tree; Heap. General tree and representations.

Sorting Algorithms

Study different sorting techniques, for example insertion sort, heapsort, merge sort, quicksort, and radix sort; Comparison of the performance and complexity of the sorting algorithms.

Hash Tables

Design of hash functions; Collision resolution and overflow handling; Algorithms for search, insert and delete operations; Performance analysis.

Depending on the students' level and progress, we may also cover the following topics (optional):

Graph representation

Graph representation and basic graph operation algorithms

Brief introduction to general algorithm design techniques

Alternative implementation using dynamic programming; basic introduction to greedy algorithm design technique.

Reading List

Compulsory Readings

	Title
1	Nil

Additional Readings

	Title
1	Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein: Introduction to Algorithms, MIT Press
2	D. S. Malik : C++ Programming Program Design Including Data Structures, 6th ed. (Cengage Learning 2013)
3	http://www.cplusplus.com/