

CS3334: DATA STRUCTURES

Effective Term

Semester A 2025/26

Part I Course Overview

Course Title

Data Structures

Subject Code

CS - Computer Science

Course Number

3334

Academic Unit

Computer Science (CS)

College/School

College of Computing (CC)

Course Duration

One Semester

Credit Units

3

Level

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

Medium of Instruction

English

Medium of Assessment

English

Prerequisites

CS2310 Computer Programming or CS2311 Computer Programming or CS2312 Problem Solving and Programming or equivalent

Precursors

Nil

Equivalent Courses

Nil

Exclusive Courses

CS2334 Data Structures for Data Science

Part II Course Details

Abstract

This course aims to provide students an appreciation to the fundamentals of computer science. Models and applications of data structures including heaps, search trees, hash tables and disjoint sets are introduced and evaluated. Mathematical tools for analysis of algorithms and data structures are discussed and applied. Students are given the opportunity to develop and implement applications of the data structures and their derivatives.

Course Intended Learning Outcomes (CILOs)

CILOs		Weighting (if DEC-A1 app.)		DEC-A2	DEC-A3
1	Implement common data structures and algorithms.			x	
2	Analyse efficiency and correctness of algorithms using mathematical techniques.			x	
3	Evaluate and compare similar data structures and algorithms.		x	x	
4	Design and apply appropriate data structures to solve problems.		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.

Learning and Teaching Activities (LTAs)

LTAs		Brief Description	CILO No.	Hours/week (if applicable)
1	Lecture	Students will explain key concepts about algorithms and data structures for searching, indexing, sorting, manipulating data.	1, 2, 3, 4	3 hours/week
2	Tutorial	Students will work on hands-on exercises and labs related to the key concepts taught in lectures.	1, 2, 3, 4	8 hours/semester
3	Assignments	Students will be required to do programming and analysis tasks.	1, 2, 3, 4	After Class

Assessment Tasks / Activities (ATs)

	ATs	CILO No.	Weighting (%)	Remarks ("- for nil entry)	Allow Use of GenAI?
1	Assignments	1, 2, 3, 4	25	Students are required to work on assignments at least once every four weeks. 1. Students may use GenAI tools to assist with understanding concepts or generating ideas, but all submissions must reflect their own understanding and effort. Directly copying and pasting answers generated by GenAI into assignments is strictly prohibited. 2. Uploading entire assignment questions, datasets, or other course materials directly to GenAI tools to obtain answers is not allowed. 3. If significant assistance is obtained from GenAI tools, students must clearly document the nature and scope of its use in their submissions.	Yes
2	Quiz	1, 2, 3, 4	15	-	No

Continuous Assessment (%)

40

Examination (%)

60

Examination Duration (Hours)

2

Minimum Continuous Assessment Passing Requirement (%)

40

Minimum Examination Passing Requirement (%)

30

Additional Information for ATs

For a student to pass the course, at least 40% of the maximum mark for the continuous assessment and 30% of the maximum mark for the examination must be obtained.

Assessment Rubrics (AR)

Assessment Task

Assignments

Criterion

CAPACITY for DIRECTED LEARNING to understand the concepts and implementation of key data structures and algorithms

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

Mid-term and Final exams

Criterion

ABILITY to apply the knowledge about the data structures and algorithms taught in the lectures and tutorials

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

Program correctness. Complexities of programs: notation, average and worst case analysis, complexities of common programming constructs. Sorting algorithms: merge sort, heap sort, quicksort, bucket sort. Algorithms for order statistics. Abstract data types: stacks, queues, heaps. Balanced search trees: AVL trees, red-black trees, B-trees. Game Trees. Hash tables. Disjoint sets. Graphs.

Syllabus:

- Program correctness and complexities
Techniques for proving program correctness, e.g., loop invariant and induction. Asymptotic notations for program complexities. Summation and recurrence formulas. Complexities of common programming constructs, e.g., loops and recursive programs. Average and worst case analysis.
- Sorting algorithms
Selected sorting algorithms, such as merge sort, heap sort, quicksort, bucket sort, radix sort, as examples to illustrate the previous concepts and analysis techniques. Algorithms for order statistics.
- Review of abstract data types
Principles of abstract data types. Examples: stacks, queues, heaps.
- Search trees
Binary search trees. Balanced search trees: AVL trees, splay trees, red-black trees, B-trees.
- Game trees
Minimax Rules. Alpha-Beta Pruning. Reducing searching depth.
- Hash tables
Hash functions. Collision resolution. Rehashing
- Disjoint Set
Disjoint set operations. Path compression. Ackermann's function.
- Graphs
Adjacency Matrix, Depth First Search, Breadth First Search
- Selected advanced topics in Data Structures

Reading List

Compulsory Readings

Title	
1	Cormen T., Leiserson C., Rivest R. and Stein C. (2009). Introduction to Algorithms. MIT Press, 3rd edition

Additional Readings

Title	
1	J. Lewis, J. Chase (2013). Java Software Structures: Designing and Using Data Structures. Pearson, 4th edition.
2	Y. D. Liang (2013). Introduction to Java™ Programming Comprehensive Version. Pearson, 9th edition.