# CS2334: DATA STRUCTURES FOR DATA SCIENCE

**Effective Term**

Semester A 2023/24

## Part I Course Overview

**Course Title**

Data Structures for Data Science

**Subject Code**

CS - Computer Science

**Course Number**

2334

**Academic Unit**

Computer Science (CS)

**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**

CS1315 Introduction to Computer Programming or
CS2311 Computer Programming or
CS2315 Computer Programming or equivalent

**Precursors**

Nil

**Equivalent Courses**

Nil

**Exclusive Courses**

CS3334 Data Structures

# 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 app.) | DEC-A1 | 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.

### Teaching and Learning Activities (TLAs)

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

**Assessment Tasks / Activities (ATs)**

| | ATs | CILO No. | Weighting (%) | Remarks (e.g. Parameter for GenAI use) |
|---|---|---|---|---|
| 1 | Assignments | 1, 2, 3, 4 | 25 | Students are required to work on assignments at least once every four weeks |
| 2 | Quiz | 1, 2, 3, 4 | 15 | |

**Continuous Assessment (%)**

40

**Examination (%)**

60

**Examination Duration (Hours)**

2

**Additional Information for ATs**

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

**Assessment Rubrics (AR)**

**Assessment Task**

1. 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**

2. 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, 2-3 trees, B-trees. Hash tables. Merge-find sets.

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, graphs.
- Search trees
  Binary search trees. Balanced search trees: AVL trees, red-black trees, B-trees.
- Hash tables
  Open addressing. Hash functions. Collision resolution.
- Disjoint sets
  Disjoint set operations. Path compression. Ackermann’s function.

### Reading List

### Compulsory Readings

| | Title |
|---|---|
| 1 | Cormen T., Leiserson C., Rivest R. and Stein C. (2022). Introduction to Algorithms. MIT Press, 4th edition |

### Additional Readings

| | Title |
|---|---|
| 1 | Weiss M. (2013). Data Structures & Algorithm Analysis in C++. Addison Wesley, 4th edition. |