COS30008

Data Structures and Patterns
Semester 1, 2025

COS30008

Convener:	Dr Markus Lumpe
Lecture:	Thursday 14:30-16:30 (ATC101)
Labs:	Monday 08:30, 10:30, 12:30 (BA405), no classes on Labour Day
	Tuesday 08:30, 10:30, 12:30, 14:30, 16:30 (EN603)
	Wednesday 08:30, 10:30, 12:30, 14:30 (EN603)
	Thursday 08:30, 10:30 (EN207)
Grading:	Problem sets (4), mid-term, final exam
Assessments:	self-guided study projects with specific deadlines

WHY STUDY DATA STRUCTURES?

"Smart data structures and dumb code work a lot better than the other way around."

Eric S. Raymond: The Cathedral and the Bazaar

SUBJECT AIMS

- How can a given problem be effectively expressed?
- What are suitable data representations for specifying computational processes?
- What is the impact of data and its representation on time and space consumption?
- What are the reoccurring structural artifacts in software, and how can we identify them to facilitate problem-solving?

LEARNING OBJECTIVES

1. Solve problems using object-oriented design and implementation techniques.

2. Document the impact of data structures on algorithms.

3. Interpret the trade-offs and issues of various data structures for a given problem in the design, implementation, and application.

4. Summarize the purpose of data structures and design patterns and compare their strengths and weaknesses concerning resource consumption.

5. Evaluate algorithm designs and perform best, average, and worst-case analyses.

6. Design, construct, and compose software solutions using behavioral, creational, and structural design patterns.

7. Create a plan and produce a complex software solution in an industry-based context using multiple non-trivial data structures and algorithms and relating them to solution alternatives.

GENERAL MARKING CRITERIA

- Program code will be validated by automated test procedures whenever possible.
- Code must compile. If the code for a given task does not compile, then this task attracts a penalty.
- Code must be free of errors. Code is checked for
  - logical errors
  - memory leaks
  - code analysis issues
  - suitability
- Tasks must be completed within the time allotted.
- Program code must adhere to given specifications and meet expected patterns.
- Written responses to questions or tasks must address the problem and follow the suggested solution approach.
- Submission and quality requirements are available on the COS30008 Canvas page.

GENERATIVE AI USE

- Generative AI has become almost ubiquitous. Many tools and frameworks incorporate generative AI to facilitate user tasks. It can also assist in learning and decision-making. A recent study revealed that generative AI can diagnose health issues with an accuracy of ninety percent, greatly outperforming GPs.

- However, the use of generative AI can easily conflict with academic integrity. Generative AI reproduces and aggregates existing knowledge. Generative AI does not create new knowledge per se. The use of generative AI must be aligned with industry standards.

- In this unit, you may use generative AI in the design phase of your solution and for editing textual responses to questions. Verbatim copies of AI-generated code or text are not permitted.

- Every student has to produce original work. Generative AI can inform activities, but submitted work must reflect the capacity and ability of a student to create a working product.

OVERVIEW

The following gives a tentative list of topics not necessarily in the order in which they will be covered in the subject:

- Introduction
- Sets, Arrays, Indexers, and Iterators
- Basic Data Structures and Patterns
- Abstract Data Types and Data Representation
- One-Dimensional Data Structures
- Hierarchical Data Structures
- Algorithmic Patterns and Problem Solvers