

# COMP 2030 DATA STRUCTURES AND ALGORITHMS (ADVANCED)

Credit Points 10

**Coordinator** Dongmo Zhang ([https://directory.westernsydney.edu.au/search/name/Dongmo Zhang/](https://directory.westernsydney.edu.au/search/name/Dongmo%20Zhang/))

**Description** This subject introduces students to fundamental data structures and algorithms used in computing related courses. The material covered forms the basis for further studies in programming and software engineering in later subjects and for further training in programming skills. The subject focuses on the ideas of data abstraction and algorithm efficiency. The issues of computational complexity of algorithms are addressed throughout the semester. The topics covered include the fundamental abstract data types (lists, stacks, queues, trees, hash tables, graphs), recursion, complexity of algorithms, sorting and searching algorithms, binary search trees and graphs. As a subject designed for advanced computer science students, the students will conduct an advanced project on top of the assignment tasks for further training of their programming skills.

**School** Computer, Data & Math Sciences

**Discipline** Data Structures

**Student Contribution Band** HECS Band 2 10cp

Check your fees via the Fees ([https://www.westernsydney.edu.au/currentstudents/current\\_students/fees/](https://www.westernsydney.edu.au/currentstudents/current_students/fees/)) page.

**Level** Undergraduate Level 2 subject

**Pre-requisite(s)** COMP2014

## Assumed Knowledge

A good understanding of the concepts in Object-Oriented Programming, such as data abstraction, encapsulation, inheritance, and polymorphism.

Knowledge of operator overloading and class template.

Solid programming skills in one of the general-purpose programming languages, such as C, C++, Java and Python.

## Learning Outcomes

After successful completion of this Subject, students will be able to:

1. Select appropriate data structures to solve moderately complex programming problems.
2. Discuss time and space tradeoffs among different data structures that could be used to represent specific information.
3. Perform time-complexity analysis against multiple implementations of various abstract data types.
4. Write programs using the fundamental data structures, including linked list, stack, queue, hash table, binary search tree, heap and graph.
5. Apply or design efficient and effective algorithms to solve moderately complex programming problems.

## Subject Content

1. Basic concepts: abstract data types, big-O concept and complexity analysis.

2. Stacks and queues: ADT specification, implementation strategies and applications.
3. Variations of linked lists: Doubly linked lists and circular lists.
4. Recursion: recursive functions and divide-and-conquer approach.
5. Trees: Binary trees, binary search trees, AVL trees, and heaps.
6. Graphs: Adjacency matrix and adjacency list implementations, depth-first search, breadth-first search, and minimum spanning tree algorithms.
7. Searching: Sequential search, binary search and hashing.
8. Elementary sorting algorithms: insertion sort, selection sort, and bubble sort.
9. Advanced sorting algorithms: quick sort, heap sort, and shell sort.
10. Typical AI algorithms: backtracking, heuristic search, Monte-Carlo tree search and so on

## Assessment

The following table summarises the standard assessment tasks for this subject. Please note this is a guide only. Assessment tasks are regularly updated, where there is a difference your Learning Guide takes precedence.

Type	Length	Percent	Threshold	Individual/Group Task
Practical	2 hours (each 14 practical)		N	Individual
Quiz	Approximately 6 30 minutes each x 10		N	Individual
Short Answer	Around 600-800 lines of code and up to 4 A4 pages of algorithm description and analysis for each assignment	20	N	Individual
Applied Project	Around 300 to 1000 lines of code in the same or different programming language of the base assignment code	10	N	Individual
Final Exam	Two-hour	50	N	Individual

## Prescribed Texts

Drozdek, A. (2013). Data structures and algorithms in C++ (4th ed.). Cengage Learning

## Teaching Periods

## Autumn (2024)

### Penrith (Kingswood)

#### On-site

**Subject Contact** Dongmo Zhang ([https://directory.westernsydney.edu.au/search/name/Dongmo Zhang/](https://directory.westernsydney.edu.au/search/name/Dongmo%20Zhang/))

View timetable ([https://classregistration.westernsydney.edu.au/even/timetable/?subject\\_code=COMP2030\\_24-AUT\\_KW\\_1#subjects](https://classregistration.westernsydney.edu.au/even/timetable/?subject_code=COMP2030_24-AUT_KW_1#subjects))

## **Parramatta - Victoria Rd**

### **On-site**

**Subject Contact** Dongmo Zhang (<https://directory.westernsydney.edu.au/search/name/Dongmo%20Zhang/>)

View timetable ([https://classregistration.westernsydney.edu.au/even/timetable/?subject\\_code=COMP2030\\_24-AUT\\_PS\\_1#subjects](https://classregistration.westernsydney.edu.au/even/timetable/?subject_code=COMP2030_24-AUT_PS_1#subjects))