Course Outline

Contents

- Course Details (https://webcms3.cse.unsw.edu.au/COMP9024/18x1/outline#details)
- Course Summary (https://webcms3.cse.unsw.edu.au/COMP9024/18x1/outline#summary)
- Assumed Knowledge (https://webcms3.cse.unsw.edu.au/COMP9024/18x1/outline#background)
- Student Learning Outcomes (https://webcms3.cse.unsw.edu.au/COMP9024/18x1/outline#outcomes)
- Teaching Strategies (https://webcms3.cse.unsw.edu.au/COMP9024/18x1/outline#strategies)
- Teaching Rationale (https://webcms3.cse.unsw.edu.au/COMP9024/18x1/outline#rationale)
- Student Conduct (https://webcms3.cse.unsw.edu.au/COMP9024/18x1/outline#conduct)
- Academic Honesty and Plagiarism (https://webcms3.cse.unsw.edu.au/COMP9024/18x1/outline#plagiarism)
- Assessment (https://webcms3.cse.unsw.edu.au/COMP9024/18x1/outline#assessment)
- Course Schedule (https://webcms3.cse.unsw.edu.au/COMP9024/18x1/outline#schedule)
- Resources for Students (https://webcms3.cse.unsw.edu.au/COMP9024/18x1/outline#resources)
- Course Evaluation and Development (https://webcms3.cse.unsw.edu.au/COMP9024/18x1/outline#evaluation)

Course Details

Course Code	COMP9024
Course Title	Data Structures and Algorithms
Convenor	Ashesh Mahidadia
Lectures	Tuesday 18:00-21:00 in CLB 8
Consultations	2pm to 5pm on Friday
Units of Credit	6
Course Website	http://www.cse.unsw.edu.au/~cs9024 (http://www.cse.unsw.edu.au/~cs9024)
Handbook Entry	http://www.handbook.unsw.edu.au/postgraduate/courses/current/COMP9024.html (http://www.handbook.unsw.edu.au/postgraduate/courses/current/COMP9024.html)

Course Summary

Data structures are about how data is stored inside a computer for effective and efficient use. An algorithm is a step-by-step process for solving a problem within a finite amount of space and time. Data structures and algorithms are not only important in software design, but also in hardware design. Being proficient in data structures and algorithms are essential for good software developers, hardware developers, and system architects.

The actual content is taken from a list of subjects that constitute the basis of the tool box of every serious practitioner of computing: data types and data structures, abstract data types, dynamic data structures, analysis of algorithms and a variety of fundamental algorithms for graphs, trees and text processing.

Assumed Knowledge

Before commencing this course, students should

- be able to design, implement and test programs written in a procedural language;
- know how to represent data with linked lists, stacks, queues, heaps and binary trees;
- be able to implement sorting algorithms.

These are assumed to have been acquired in the course COMP9021.

Student Learning Outcomes

After successfully completing this course, students will know fundamental data structures and algorithms, and they will be able to reason about their applicability, effectiveness and efficiency.

This course contributes to the development of the following graduate capabilities:

Graduate Capability	Acquired in
scholarship: understanding of their discipline in its interdisciplinary context	lectures

scholarship: capable of independent and collaborative enquiry	Homework and assignments
scholarship: rigorous in their analysis, critique, and reflection	Homework and assignments
scholarship: able to apply their knowledge and skills to solving problems	Homework and assignments
scholarship: capable of effective communication	forum
scholarship: information literate	lectures, homework, and assignments
scholarship: digitally literate	lectures, homework, and assignments
professionalism: capable of independent, self-directed practice	Homework and assignments
professionalism: capable of operating within an agreed Code of Practice	all course-work, by doing it yourself
global citizens: culturally aware and capable of respecting diversity and acting in socially just/responsible ways	interaction with your fellow students

Teaching Strategies

- **Lectures** introduce concepts and show examples
- **Homework** reinforces concepts and allows students to solve problems by applying the knowledge learned
- **Assignments** further reinforce concepts and allow students to solve larger problems

Teaching Rationale

Lectures will include exercises where we examine the practice of formulating and proving mathematical properties of relevance to Computer Science. Homework aims to deepen analysis and understanding via additional problems. Assignments give you the chance to practice what you have learnt on larger problems.

Student Conduct

The **Student Code of Conduct** (Information (https://student.unsw.edu.au/conduct) , Policy (https://www.gs.unsw.edu.au/policy/documents/studentcodepolicy.pdf)) sets out what the University expects from students as members of the UNSW community. As well as the learning, teaching and research environment, the University aims to provide an environment that enables students to achieve their full potential and to provide an experience consistent with the University's values and guiding principles. A condition of enrolment is that students *inform themselves* of the University's rules and policies affecting them, and conduct themselves accordingly.

In particular, students have the responsibility to observe standards of equity and respect in dealing with every member of the University community. This applies to all activities on UNSW premises and all external activities related to study and research. This includes behaviour in person as well as behaviour on social media, for example Facebook groups set up for the purpose of discussing UNSW courses or course work. Behaviour that is considered in breach of the Student Code Policy as discriminatory, sexually inappropriate, bullying, harassing, invading another one's privacy or causing any person to fear for their personal safety is serious misconduct and can lead to severe penalties, including suspension or exclusion from UNSW.

If you have any concerns, you may raise them with your lecturer, or approach the School Ethics Officer (mailto:ethics-officer@cse.unsw.edu.au) , Grievance Officer (mailto:grievance-officer@cse.unsw.edu.au) , or one of the student representatives.

Academic Honesty and Plagiarism

Plagiarism is defined as (https://student.unsw.edu.au/plagiarism) using the words or ideas of others and presenting them as your own. UNSW and CSE treat plagiarism as academic misconduct, which means that it carries penalties as severe as being excluded from further study at UNSW. There are several on-line sources to help you understand what plagiarism is and how it is dealt with at UNSW:

- Plagiarism and Academic Integrity (https://student.unsw.edu.au/plagiarism)
- UNSW Plagiarism Procedure (https://www.gs.unsw.edu.au/policy/documents/plagiarismprocedure.pdf)

Make sure that you read and understand these. Ignorance is not accepted as an excuse for plagiarism. In particular, you are also responsible that your assignment files are not accessible by anyone but you by setting the correct permissions in your CSE directory and code repository, if using. Note also that plagiarism includes paying or asking another person to do a piece of work for you and then submitting it as your own work.

UNSW has an ongoing commitment to fostering a culture of learning informed by academic integrity. All UNSW staff and students have a responsibility to adhere to this principle of academic integrity. Plagiarism undermines academic integrity and is not tolerated at UNSW. Plagiarism at UNSW is defined as using the words or ideas of others and passing them off as your own.

If you haven't done so yet, please take the time to read the full text of

- UNSW's policy regarding academic honesty and plagiarism (https://student.unsw.edu.au/plagiarism)

The pages below describe the policies and procedures in more detail:

- Student Code Policy (https://www.gs.unsw.edu.au/policy/documents/studentcodepolicy.pdf)
- Student Misconduct Procedure (https://www.gs.unsw.edu.au/policy/documents/studentmisconductprocedures.pdf)

- Plagiarism Policy Statement (https://www.gs.unsw.edu.au/policy/documents/plagiarismpolicy.pdf)
- Plagiarism Procedure (https://www.gs.unsw.edu.au/policy/documents/plagiarismprocedure.pdf)

You should also read the following page which describes your rights and responsibilities in the CSE context:

- Essential Advice for CSE Students (https://www.engineering.unsw.edu.au/computer-science-engineering/about-us/organisational-structure/student-services/policies/essential-advice-for-cse-students)

Assessment

Component	**Maximum Mark**
Assignment 1 (due in Week 5)	10
Assignment 2 (due in Week 8)	10
Assignment 3 (due in Week 10)	10
Assignment 4 (due in Week 13)	10
Final Exam	60

Your final overall mark will be the sum of your marks for each component.

To pass the course, your final overall mark must be 50 or higher **and** the final exam must be 27 or higher. Students who do not meet these requirements but achieve an overall mark ≥45 can sit the supplementary exam, in which they have to achieve a mark ≥50 to pass with a final mark of 50.

Course Schedule

Elementary data structures and algorithms in C	week 1
Abstract data types and dynamic data structures	week 2-3
Analysis of algorithms	week 4
Hash Tables	Week 5
Priority queues and disjoint set data structures	Week 6
Graph algorithms	week 7-8
Tree algorithms	week 9-10
Text processing algorithms	week 11
Randomised algorithms	week 12

Resources for Students

The recommended textbooks associated with this course are

- Robert Sedgewick, *Algorithms in C, Parts 1—4* 3rd edition, Addison Wesley, 1998.
- Robert Sedgewick, *Algorithms in C, Part 5* 3rd edition, Addison Wesley, 2002.

The following introduction to the C programming language is recommended as a supplementary textbook:

- Alistair Moffat, *Programming, Problem Solving, and Abstraction with C* , 5th edition, Pearson, 2003.

Course Evaluation and Development

This course is being continuously improved and we will conduct a survey through UNSW's myExperience process at the end of session to obtain feedback on the quality of the various course components. Your participation in the survey will be greatly appreciated, especially since this is the first time that this course uses the C programming language and is run in the new format outlined above. Students are also strongly encouraged to provide informal feedback during the session, and to notify the lecturer-in-charge of any problems as soon as they arise.

Resource created 4 years ago (Friday 23 February 2018, 03:30:37 PM), last modified 4 years ago (Tuesday 27 February 2018, 10:23:43 PM).

Comments (/COMP9024/18s1/forums/search?forum_choice=resource/15085) (/COMP9024/18s1/forums/resource/15085)

There are no comments yet.