UNSW Course Outline

COMP2521 Data Structures and Algorithms - 2024

Published on the 28 May 2024

General Course Information

Course Code : COMP2521
Year : 2024
Term : Term 2
Teaching Period : T2
Is a multi-term course? : No
Faculty : Faculty of Engineering
Academic Unit : School of Computer Science and Engineering
Delivery Mode : In Person
Delivery Format : Standard
Delivery Location : Kensington
Campus : Sydney
Study Level : Undergraduate
Units of Credit : 6

Useful Links

Handbook Class Timetable

Course Details & Outcomes

Course Description

The aim of this course is to get you to **think like a computer scientist** . This certainly sounds like a noble goal... but what does it really mean? How does a *scientist* , let alone a computer scientist, actually think?

What many types of scientists try to do is understand natural systems and processes: a geologist, for example, tries to understand the structure of the earth; a biologist tries to understand living organisms; a chemist tries to understand materials and reactions, and so on.

Computer scientists don't, as the name might suggest, simply try to understand the structure and behaviour of computers, but are more concerned with understanding software systems (and the interaction between the software and the hardware on which it runs). Also, unlike other scientists, computer scientists frequently build the objects that they study.

The goal of this course is to deepen your understanding of data structures and algorithms and how these can be employed effectively in the design of software systems. It is an important course in covering a range of core data structures and algorithms that will be used in context in later courses. You explore these ideas in lectures, tutorials, lab exercises, quizzes and assignments. Assessment involves lab exercises, quizzes, assignments and a final exam involving both practice and theory. At the end of the course, we want you to be a solid programmer, with knowledge of a range of useful data structures and programming techniques, and ready to continue with further specialised studies in computing.

Topics

This course provides an introduction to the structure, analysis and usage of a range of fundamental data types and the core algorithms that operate on them. Key topics are:

- Recursion
- Analysis of algorithms
- Abstract data types
- Binary search trees
- Balanced search trees
- Graphs
- Sorting algorithms
- Heaps
- Hashing
- Tries

Course Aims

During this course, we'll be looking at ways of creating, analysing and understanding software. Ultimately, you should be able to answer the question, "i**s this piece of software any good?**" and be able to provide sound reasons to justify your answer.

This course follows on from introductory C programming courses: COMP1511, COMP1917, or

COMP1921. We cover additional aspects of the C programming language that were not covered in those courses, and also look at some programming tools which were not covered (in detail) earlier. However, this course is not simply a second C programming course: the focus is on the ideas and abstractions behind the data structures and algorithms that are used.

COMP2521 is a critical course in the study of computing at UNSW, since it deals with many concepts that are central to future studies in the area. Whether you are studying Computer Science, Software Engineering, Bioinformatics, Computer Engineering, or even a discipline outside the realm of computing, understanding a range of algorithms and data structures and how to use them will make you a much more effective computing problem solver in the future.

Relationship to Other Courses

The prerequisite of this course is COMP1511 (or COMP1911 with a bridging course), and therefore students are expected to be familiar with the C programming language.

This course is a prerequisite of many higher-level computing courses, and it is therefore recommended that this course be taken early in any computer science or related degree.

Course Learning Outcomes

Course Learning Outcomes
CLO1 : Describe and analyse fundamental data structures and algorithms
CLO2 : Analyse the performance characteristics of algorithms
CLO3 : Measure the performance behaviour of programs
CLO4 : Choose/develop an appropriate data structure for a given problem
CLO5 : Choose/develop appropriate algorithms to manipulate chosen data structures
CLO6 : Reason about the effectiveness of data structures and algorithms for solving a given problem
CLO7 : Package a set of data structures and algorithms as an abstract data type
CLO8 : Develop and maintain software systems in C that contain thousands of lines of code

Course Learning Outcomes	Assessment Item
CLO1 : Describe and analyse fundamental data structures and algorithms	• Labs • Quizzes • Final Exam
CLO2 : Analyse the performance characteristics of algorithms	• Quizzes • Final Exam
CLO3 : Measure the performance behaviour of programs	• Labs • Quizzes • Final Exam
CLO4 : Choose/develop an appropriate data structure for a given problem	• Assignment 1 - Trees • Assignment 2 - Graphs • Labs • Quizzes • Final Exam
CLO5 : Choose/develop appropriate algorithms to manipulate chosen data structures	• Assignment 1 - Trees • Assignment 2 - Graphs • Labs • Quizzes • Final Exam
CLO6 : Reason about the effectiveness of data structures and algorithms for solving a given problem	• Quizzes • Final Exam
CLO7 : Package a set of data structures and algorithms as an abstract data type	• Assignment 1 - Trees • Assignment 2 - Graphs • Quizzes • Final Exam
CLO8 : Develop and maintain software systems in C that contain thousands of lines of code	• Assignment 1 - Trees • Assignment 2 - Graphs • Final Exam

Learning and Teaching Technologies

Moodle - Learning Management System | Echo 360 | EdStem | YouTube Live | Zoom

Learning and Teaching in this course

Lectures

Each week, there will be four hours of lectures during which theory and practical demonstrations will be presented. Lectures convey a small amount of information about the course content, but their main aim is to try to stimulate you to think about concepts and techniques.

Lectures will run at the following times:

- Monday 11am-1pm in Clancy Auditorium
- Thursday 11am-1pm online

All lectures will be recorded and made available on YouTube. We will also endeavour to livestream lectures to YouTube.

Tutorial/Lab Classes

Every week starting from Week 1, you will be expected to attend a three hour tutorial/lab class to clarify ideas from lectures and work through lab exercises. Classes begin with a 1 hour tutorial, followed by a 2 hour lab.

Most classes will be face-to-face, but there will be some online classes.

Tutorials

Tutorials aim to clarify ideas from lectures and to get you to think about design/analysis issues. The aim of the class is *not* to simply get the tutor to give you the answers; the aim is to focus on just one or two of the exercises and work through them in detail, discussing as many aspects, alternative approaches, fine details, etc. as possible. You must be active and ask questions in tutorials.

You should make sure that you use this time effectively by **examining in advance** the material to be covered in each week's tutorial. This means that you are coming to class prepared to ask any questions that you may have, and generally participate in class by offering suggestions - this will ensure that you get the most possible out of the tutorial. Your tutors are there to help you clear

up any misunderstandings or to understand topics in more depth. The tutorial questions will be linked to the class webpage in the week before each tutorial. There are no marks for tutorial attendance, however, it is your chance to have all your questions answered.

Assessments

Assessment Structure

Assessment Item	Weight	Relevant Dates
Labs Assessment Format: Individual	15%	Start Date: Start of each week (except week 6 and 10) Due Date: Monday the following week
Quizzes Assessment Format: Individual	10%	Start Date: Start of each week (except week 6 and 10) Due Date: Monday the following week
Assignment 1 - Trees Assessment Format: Individual	15%	Start Date: Week 4 Due Date: Monday Week 7
Assignment 2 - Graphs Assessment Format: Individual	15%	Start Date: Week 7 Due Date: Friday Week 10
Final Exam Assessment Format: Individual	45%	Due Date: During Exam Period.

Assessment Details

Labs

Assessment Overview

Labs aim to give you practice in problem solving and program development. Each week, there will be a few exercises to work on. These exercises will be released in the week preceding the lab class. Labs are to be completed individually.

Each lab exercise is worth 5 marks. Marks for each lab consist of an automarking component, which is based on the correctness of the code, and a handmarking component, which is based on other aspects such as style and complexity analysis. The weightings of these components vary depending on the lab. Specific details can be found in the specification for each lab.

Labs that have an automarking component will be automarked after submissions are closed. Marks for the handmarking component can be obtained by showing your work to your tutor during your lab session, within two weeks of the lab. You must show the work that you submitted, not a modified version of your work or someone else's work. Your tutor will provide feedback on your approach to the problem and on the style of your solution.

Your final lab mark will be made up of the best 7 labs, with the lowest mark being discarded. For example, if your scores for the labs are 2/5, 3/5, 3/5, 4/5, 4/5, 4/5, 5/5 and 5/5, then the 2/5 would be discarded, and your final percentage for the labs would be 28/35 = 80%.

Course Learning Outcomes

- CLO1 : Describe and analyse fundamental data structures and algorithms
- CLO3 : Measure the performance behaviour of programs
- CLO4 : Choose/develop an appropriate data structure for a given problem
- CLO5 : Choose/develop appropriate algorithms to manipulate chosen data structures

Detailed Assessment Description

See the course website for details.

Submission notes

Submitted via give

Quizzes

Assessment Overview

At the beginning of most weeks, a quiz will be released on WebCMS which tests your understanding of the lecture content from that week and the week prior. Quizzes are to be completed individually in your own time.

Each quiz will be marked out of 8. Your final quiz mark will be made up of the best 7 quizzes, with the lowest mark being discarded. For example, if your scores for the quizzes were 5/8, 5/8, 6/8, 7/8, 7/8, 7/8, 7/8 and 8/8, then one of the 5/8's would be discarded, and your final percentage for the quizzes would be 47/56 = 83.9%.

Course Learning Outcomes

- CLO1 : Describe and analyse fundamental data structures and algorithms
- CLO2 : Analyse the performance characteristics of algorithms
- CLO3 : Measure the performance behaviour of programs
- CLO4 : Choose/develop an appropriate data structure for a given problem
- CLO5 : Choose/develop appropriate algorithms to manipulate chosen data structures
- CLO6 : Reason about the effectiveness of data structures and algorithms for solving a given problem
- CLO7 : Package a set of data structures and algorithms as an abstract data type

Detailed Assessment Description

See the course website for details.

Submission notes

Submitted via WebCMS

Assignment 1 - Trees

Assessment Overview

In this assignment, you develop a C program based on a tree data structure. Marks for the assignment are based primarily on automarking, with some additional marks awarded by the tutor for programming style. Standard UNSW late penalties apply.

Course Learning Outcomes

- CLO4 : Choose/develop an appropriate data structure for a given problem
- CLO5 : Choose/develop appropriate algorithms to manipulate chosen data structures
- CLO7 : Package a set of data structures and algorithms as an abstract data type
- CLO8 : Develop and maintain software systems in C that contain thousands of lines of code

Detailed Assessment Description

See the course website for details.

Submission notes

Submitted via give

Assignment 2 - Graphs

Assessment Overview

In this assignment, you develop a C program based on a tree data structure. Marks for the assignment are based primarily on automarking, with some additional marks awarded by the tutor for programming style. Standard UNSW late penalties apply.

Course Learning Outcomes

- CLO4 : Choose/develop an appropriate data structure for a given problem
- CLO5 : Choose/develop appropriate algorithms to manipulate chosen data structures
- CLO7 : Package a set of data structures and algorithms as an abstract data type
- CLO8 : Develop and maintain software systems in C that contain thousands of lines of code

Detailed Assessment Description

See the course website for details.

Submission notes

Submitted via give

Final Exam

Assessment Overview

There will be a three-hour final exam, held in the CSE labs during the exam period. The exam is closed-book, but with relevant documentation supplied.

The final exam is a summative assessment that will test your understanding of concepts covered in the course and your programming ability. The exam will contain short-answer questions and programming questions. A sample exam will be provided in Week 10 to demonstrate the format of the exam.

Course Learning Outcomes

- CLO1 : Describe and analyse fundamental data structures and algorithms
- CLO2 : Analyse the performance characteristics of algorithms
- CLO3 : Measure the performance behaviour of programs
- CLO4 : Choose/develop an appropriate data structure for a given problem
- CLO5 : Choose/develop appropriate algorithms to manipulate chosen data structures
- CLO6 : Reason about the effectiveness of data structures and algorithms for solving a given problem
- CLO7 : Package a set of data structures and algorithms as an abstract data type
- CLO8 : Develop and maintain software systems in C that contain thousands of lines of code

Detailed Assessment Description

See the course website for details.

Hurdle rules

Details about the hurdle will be made available in week 10.

General Assessment Information

In labs and assignments, any code you submit that was copied or derived from anyone other than yourself must be clearly referenced. A guide for referencing code can be found on the course website.

Grading Basis

Standard

Requirements to pass course

To pass this course, you must pass the final exam hurdle and achieve a composite mark of at least 50 out of 100.

The final exam hurdle consists of:

- at least 18/45 (40%) on the final exam
- at least 25% on each of the two components (theory and practical) of the final exam

Course Schedule

Teaching Week/Module	Activity Type	Content
Week 1 : 27 May - 2 June	Lecture	Course introduction, recursion, analysis of algorithms
Week 2 : 3 June - 9 June	Lecture	Sorting algorithms
Week 3 : 10 June - 16 June	Lecture	Abstract data types, binary search trees
Week 4 : 17 June - 23 June	Lecture	Binary search trees (continued), introduction to graphs
Week 5 : 24 June - 30 June	Lecture	Graph traversal, graph problems
Week 6 : 1 July - 7 July	Other	No classes
Week 7 : 8 July - 14 July	Lecture	Directed/weighted graphs, Dijkstra's algorithm, minimum spanning trees
Week 8 : 15 July - 21 July	Lecture	Hash tables
Week 9 : 22 July - 28 July	Lecture	Priority queues and heaps, tries
Week 10 : 29 July - 4 August	Lecture	Course review, final exam

Attendance Requirements

Students are strongly encouraged to attend all classes and review lecture recordings.

Course Resources

Prescribed Resources

Forum

COMP2521 uses the Ed forum.

The forum is a resource where students can ask questions to clarify their understanding of the course material and improve their knowledge, share helpful resources and insights, and ask for help.

Student participation is highly encouraged.

Recommended Resources

Textbooks

You are not required to purchase any textbooks. Some students have found the textbooks below to be useful in the past:

- *Algorithms in C, Parts 1-4: Fundamentals, Data Structures, Sorting, Searching* (3rd Edition) by Robert Sedgewick, published by Addison-Wesley

- *Algorithms in C, Part 5: Graph Algorithms* (3rd Edition)
  by Robert Sedgewick, published by Addison Wesley
- *Introduction to Algorithms* (Fourth Edition)
  by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein
- *The Art of Computer Programming* (Third Edition)
  by Donald E. Knuth

Course Evaluation and Development

Student feedback on the effectiveness of lectures, tutorials and labs is obtained via the myExperience survey at the end of each term. Student feedback is taken seriously, and continual improvements are made to the course based in part on this feedback. Students are strongly encouraged to let the lecturer in charge know of any problems as soon as they arise. Suggestions and criticisms will be listened to openly, and every action will be taken to correct any issue or improve the students' learning experience.

Staff Details

Position	Name	Email	Location	Phone	Availability	Equitable Learning Services Contact	Primary Contact
Administrator	COMP2521 Course Administrators					Yes	Yes
Convenor	Sim Mautner					No	No
Administrator	Kevin Luxa					No	No
	Ethan Brown					No	No

Other Useful Information

Academic Information

I. Special consideration and supplementary assessment

If you have experienced an illness or misadventure beyond your control that will interfere with your assessment performance, you are eligible to apply for Special Consideration prior to, or within 3 working days of, submitting an assessment or sitting an exam.

Please note that UNSW has a Fit to Sit rule, which means that if you sit an exam, you are declaring yourself fit enough to do so and cannot later apply for Special Consideration.

For details of applying for Special Consideration and conditions for the award of supplementary assessment, please see the information on UNSW's Special Consideration page.

II. Administrative matters and links

All students are expected to read and be familiar with UNSW guidelines and polices. In particular, students should be familiar with the following:

- Attendance
- UNSW Email Address
- Special Consideration
- Exams
- Approved Calculators
- Academic Honesty and Plagiarism
- Equitable Learning Services

III. Equity and diversity

Those students who have a disability that requires some adjustment in their teaching or learning environment are encouraged to discuss their study needs with the course convener prior to, or at the commencement of, their course, or with the Equity Officer (Disability) in the Equitable Learning Services. Issues to be discussed may include access to materials, signers or note-takers, the provision of services and additional exam and assessment arrangements. Early notification is essential to enable any necessary adjustments to be made.

IV. Professional Outcomes and Program Design

Students are able to review the relevant professional outcomes and program designs for their streams by going to the following link: https://www.unsw.edu.au/engineering/student-life/student-resources/program-design.

Note: This course outline sets out the description of classes at the date the Course Outline is published. The nature of classes may change during the Term after the Course Outline is published. Moodle or your primary learning management system (LMS) should be consulted for the up-to-date class descriptions. If there is any inconsistency in the description of activities between the University timetable and the Course Outline/Moodle/LMS, the description in the Course Outline/Moodle/LMS applies.

Academic Honesty and Plagarism

UNSW has an ongoing commitment to fostering a culture of learning informed by academic integrity. All UNSW 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.

Plagiarism is a type of intellectual theft. It can take many forms, from deliberate cheating to accidentally copying from a source without acknowledgement. UNSW has produced a website with a wealth of resources to support students to understand and avoid plagiarism, visit: student.unsw.edu.au/plagiarism. The Learning Centre assists students with understanding academic integrity and how not to plagiarise. They also hold workshops and can help students one-on-one.

You are also reminded that careful time management is an important part of study and one of the identified causes of plagiarism is poor time management. Students should allow sufficient time for research, drafting and the proper referencing of sources in preparing all assessment tasks.

Repeated plagiarism (even in first year), plagiarism after first year, or serious instances, may also be investigated under the Student Misconduct Procedures. The penalties under the procedures can include a reduction in marks, failing a course or for the most serious matters (like plagiarism in an honours thesis or contract cheating) even suspension from the university. The Student Misconduct Procedures are available here:

www.gs.unsw.edu.au/policy/documents/studentmisconductprocedures.pdf

Submission of Assessment Tasks

Work submitted late without an approved extension by the course coordinator or delegated authority is subject to a late penalty of five percent (5%) of the maximum mark possible for that assessment item, per calendar day.

The late penalty is applied per calendar day (including weekends and public holidays) that the assessment is overdue. There is no pro-rata of the late penalty for submissions made part way through a day. This is for all assessments where a penalty applies.

Work submitted after five days (120 hours) will not be accepted and a mark of zero will be awarded for that assessment item.

For some assessment items, a late penalty may not be appropriate. These will be clearly indicated in the course outline, and such assessments will receive a mark of zero if not completed by the specified date. Examples include:

- Weekly online tests or laboratory work worth a small proportion of the subject mark;
- Exams, peer feedback and team evaluation surveys;
- Online quizzes where answers are released to students on completion;
- Professional assessment tasks, where the intention is to create an authentic assessment that has an absolute submission date; and,
- Pass/Fail assessment tasks.

Faculty-specific Information

Engineering Student Support Services – The Nucleus - enrolment, progression checks, clash requests, course issues or program-related queries

Engineering Industrial Training – Industrial training questions

UNSW Study Abroad – study abroad student enquiries (for inbound students)

UNSW Exchange – student exchange enquiries (for inbound students)

UNSW Future Students – potential student enquiries e.g. admissions, fees, programs, credit transfer

Phone

(+61 2) 9385 8500 – Nucleus Student Hub

(+61 2) 9385 7661 – Engineering Industrial Training

(+61 2) 9385 3179 – UNSW Study Abroad and UNSW Exchange (for inbound students)

School Contact Information

CSE Help! - on the Ground Floor of K17

- For assistance with coursework assessments.

The Nucleus Student Hub - https://nucleus.unsw.edu.au/en/contact-us

- Course enrolment queries.

Grievance Officer - grievance-officer@cse.unsw.edu.au

- If the course convenor gives an inadequate response to a query or when the courses convenor does not respond to a query about assessment.

Student Reps - stureps@cse.unsw.edu.au

- If some aspect of a course needs urgent improvement. (e.g. Nobody responding to forum queries, cannot understand the lecturer)

You should **never** contact any of the following people directly:

- Vice Chancellor

- Pro-vice Chancellor Education (PVCE)

- Head of School

- CSE administrative staff

- CSE teaching support staff

They will simply bounce the email to one of the above, thereby creating an unnecessary level of indirection and a delay in the response.