Course Outline Public

Edit Resource (/COMP9024/19T2/resources/25282/edit)

Contents

- Course Details
- Course Summary
- Assumed Knowledge
- Student Learning Outcomes
- Teaching Strategies
- Teaching Rationale
- Student Conduct
- Academic Honesty and Plagiarism
- Assessment
- Special Consideration
- Late Submission
- Course Schedule
- Laboratory
- Consultation
- Problem Sets
- Recommended Textbooks
- Course Evaluation and Development

Course Details

Course Code	COMP9024
Course Title	Data Structures and Algorithms
Convenor	Adjunct Assoc. Prof. Albert Nymeyer
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 used to store data inside a computer. It's crucial that data can be accessed and manipulated efficiently. An algorithm is a step-by-step process for solving a problem in a finite amount of space and time. The design of an algorithm can heavily influence the design of a data structure, and vice versa. Understanding how data and algorithm must work together is essential for good software and hardware developers and system architects. The content of this course forms the basis of a tool box that every serious computer-science practitioner should have: data types and structures, abstract data types, dynamic data structures, complexity analysis and a variety of basic graph and tree processing algorithms.

Assumed Knowledge

Before commencing this course, students should

- know what fundamental data types such as characters and integers, and structures such as arrays are
- understand the concept of program memory
- be able to write and test programs written in a procedural language
- know what linked lists, stacks, queues and binary trees are
- be able to write a sorting algorithm

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	problem sets, assignments, quiz
scholarship: rigorous in their analysis, critique, and reflection	in problem sets, assignments
scholarship: able to apply their knowledge and skills to solving problems	problem sets and assignments
scholarship: capable of effective communication	forum
scholarship: information literate	lectures, problem sets, assignments
scholarship: digitally literate	lectures, problem sets, assignments
professionalism: capable of independent, self-directed practice	problem sets and assignments
professionalism: capable of operating within an agreed Code of Practice	all coursework, 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
- **Problem sets** reinforce concepts, provide additional examples and allow students to solve problems
- **Assignments** further reinforce concepts and allow students to solve larger problems
- **Mid-term examination** test understanding and skills

Teaching Rationale

Lectures will include worked programs that explain concepts as well as mathematical analyses used in performance measurement. Exercises are provided that reinforce learning and develop programming skills. Assignments are an opportunity to apply your skills to larger problems.

Be mindful of the importance of spending time on **actual** programming. The C language is particularly complex requiring attention to both high-level design and low-level detail at the same time. Most students require a whole session of practice at C programming to achieve the required level of competency.

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

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 4)	10
Mid-term online exam (in week 5)	15
Assignment 2 (due in week 10)	15
Final Exam (exam period)	60

Your final overall mark will be the sum of your marks for each component provided that you pass the final exam.

To pass the course, the sum of your marks must be 50 or higher **and** the mark for the final exam must be 25 or higher.

Special Consideration

If your work in this course is affected by unforeseen adverse circumstances, you should apply for special consideration. If your request is reasonable and your work has clearly been impacted, then:

- for an exam (or quiz), you may be offered a supplementary exam (or quiz)
- for an assignment, you may be granted an extension

Note the use of the word "may". None of the above is guaranteed. It depends on whether the circumstances clearly have impacted your ability to work.

UNSW handles special consideration requests centrally (in the Student Lifecycle division), so all special consideration requests must be submitted via the UNSW Special Consideration (https://iaro.online.unsw.edu.au/special_consideration/home.login) website. Special consideration requests must be accompanied by documentation, which will be verified by Student Lifecycle. Please do not email the course convenor directly regarding special consideration requests.

If you attend the exam, you cannot apply for special consideration afterwards (you are considered 'fit-to-sit'). If you cannot attend the exam because of illness or circumstance then you should request special consideration, with documentation, through the website within 24 hours of the exam. If your request is accepted then you will be awarded a supplementary exam. Note that a supplementary final exam is not automatically awarded if you achieve a mark close to but under 50% as some courses do at CSE. Also note that UNSW expects you to be available to sit for the supplementary final exam held after the examination period. If you are awarded a supplementary exam and do not attend then the corresponding mark will be zero. For further details on special consideration, see the website for Current Students (https://student.unsw.edu.au/special-consideration) .

Late Submission

It is better to hand in an assignment late than not at all because 1) you will get some marks if it isn't more than 4 days late and 2) much more importantly, you will learn just as much as if it had been on time, and it is this learning that will help you in other assessments such as the final exam.

The general formula for the penalties for submitting late is:

- up to 1 day (i.e. within 24 hours) the mark will be scaled by 80%
- for 2 days it is 70% (that is total scaling, not in addition to the 80%)
- for 3 it is 60%
- for 4 it is 50%

After four days the submission will be accepted but the mark will be 0%. If you think you have sound reasons to request a waiver of these rules then you should go through the special-consideration process as described above.

Course Schedule

Very approximately, the schedule of topics is:

Elementary data structures and algorithms in C	week 1-2
Abstract data types and dynamic data structures	week 3
Analysis of algorithms and randomness	week 4
Mid-term online test	**week 5**
Graph algorithms	week 6-7
Tree algorithms	week 8-10

Laboratory

A laboratory is scheduled every week, but it is voluntary. There will be a tutor available to answer questions. See the course homepage for the time and place.

Consultation

A consultation is also scheduled every week for those that wish to speak with the convenor. See the course homepage for the time and place.

Problem Sets

There is a set of exercises every week, but doing them is entirely voluntary. Exercises are both theoretical and programming in nature.

The solutions to the exercises will be revealed the following week when a new set of exercises is posted. It's really important that you spend some time trying to solve the exercises yourself before you see the solutions (that's when you learn!).

Recommended Texts

- 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 also recommended:

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

Course Evaluation and Development

This course is under constant revision, with one of the main drivers being the feedback you provide UNSW's myExperience survey at the end of session. You are also encouraged to provide informal feedback during the session, and to notify the course admin of any problems as soon as they arise.

Student feedback from 18s2 rated the course very highly and there were no particular issues. This course will handle the same topics but the lecture slides have been placed on a wiki, which is more conducive to explaining programming concepts.

Resource created about a year ago (Thursday 21 March 2019, 10:06:44 AM), last modified 11 months ago (Monday 03 June 2019, 03:20:02 PM).