UNSW Course Outline

COMP4128 Programming Challenges - 2024

Published on the 25 Aug 2024

General Course Information

Course Code : COMP4128
Year : 2024
Term : Term 3
Teaching Period : T3
Is a multi-term course? : No
Faculty : Faculty of Engineering
Academic Unit : School of Computer Science and Engineering
Delivery Mode : Multimodal
Delivery Format : Standard
Delivery Location : Kensington
Campus : Sydney
Study Level : Postgraduate, Undergraduate
Units of Credit : 6

Useful Links

Handbook Class Timetable

Course Details & Outcomes

Course Description

How do competitive programmers solve complex problems in a matter of minutes?

In this course, you will design and implement advanced algorithms to solve problems accurately

and quickly. You will discover sophisticated applications of dynamic programming, data structures, graph algorithms, mathematics and more. Most importantly, you will learn to deconstruct a problem in order to evaluate which algorithm design techniques are appropriate, combining ideas from different contexts, and address the challenges that lie in both solving the problem conceptually and producing a C/C++ program which enacts your solution. Can you rise to the challenge?

Course Aims

It is common for computing professionals to be faced with challenging problems that require both algorithmic thinking and skillful programming to solve. The aim of this course is to train students in the implementation of solutions of such problems, focusing on accuracy, efficiency and speed.

Relationship to Other Courses

This course extends COMP3121/9101/3821/9801 to include applications of problem-solving techniques, in order to code solutions to problems, particularly in the style of programming contests. As such, the course trains students to solve problems in this field across a variety of topics, including under time constraints.

- The prerequisite for this course is COMP3821/9801 (or COMP3121/9101 and 75 WAM).
- Students are assumed to have a thorough understanding of basic algorithms and data structures, as covered in COMP1511/9021 and COMP2521/9024.
- Students will benefit from having some familiarity with more advanced algorithms and data structures, as covered in COMP3121/9101 or COMP3821/9801, although much of the material is reintroduced.
- Students are expected to be proficient in C or C++, as all coding in this course will be in C++. A detailed understanding of C++ classes from an object-oriented perspective is not required. However, students must be able to write, test and debug relatively short programs in C, and familiarity with the C++ Standard Template Library will be developed early in the term.

Course Learning Outcomes

Course Learning Outcomes
CLO1 : Apply the C++ programming language to solve programming problems
CLO2 : Adapt templated data structures and algorithms to meet the requirements of complex problems
CLO3 : Design and implement algorithmic solutions to complex problems, particularly under short time constraints
CLO4 : Apply algorithm design principles to evaluate programs in terms of correctness and running time

Course Learning Outcomes	Assessment Item
CLO1 : Apply the C++ programming language to solve programming problems	• Final Exam • Problem Sets • Contests
CLO2 : Adapt templated data structures and algorithms to meet the requirements of complex problems	• Final Exam • Problem Sets • Contests
CLO3 : Design and implement algorithmic solutions to complex problems, particularly under short time constraints	• Final Exam • Problem Sets • Contests
CLO4 : Apply algorithm design principles to evaluate programs in terms of correctness and running time	• Final Exam • Problem Sets • Contests

Learning and Teaching Technologies

Moodle - Learning Management System | Blackboard Collaborate | EdStem | Echo 360

Learning and Teaching in this course

Lectures will be delivered in hybrid mode; you can attend in person or watch the Echo360 live stream, which you can access through Moodle. Echo360 will also give you access to recordings of past lectures.

You can find the full workshop and lab timetable here. The online classes will be delivered on Blackboard Collaborate, which you can also access through Moodle.

Please email me if you would like a consultation (likely online) outside of the specified hours.

I will hold additional consultations during the exam period, schedule TBA.

Assessments

Assessment Structure

Assessment Item	Weight	Relevant Dates
Final Exam Assessment Format: Individual	34%	Start Date: TBA during exam period Due Date: TBA during exam period
Problem Sets Assessment Format: Individual	48%	Start Date: Not Applicable Due Date: End of exam period
Contests Assessment Format: Individual	18%	Start Date: Weeks 1, 5 and 9 (TBC) Due Date: Weeks 1, 5 and 9 (TBC)

Assessment Details

Final Exam

Assessment Overview

The final exam is held during the UNSW exam period and is a long contest (approximately 5 hours).

This contest consists of programming problems on the entirety of course material, with immediate feedback for each submission.

Submissions are automatically marked against a batch of test cases fitting the specifications provided to students. Students receive feedback of 'Accepted', 'Wrong Answer', 'Time Limit Exceeded', 'Runtime Error' and so on for each attempt, and have unlimited attempts at each problem to get the 'Accepted' verdict. Marks are again awarded for the number of problems where the 'Accepted' verdict is received, including subtasks.

Course Learning Outcomes

- CLO1 : Apply the C++ programming language to solve programming problems
- CLO2 : Adapt templated data structures and algorithms to meet the requirements of complex problems
- CLO3 : Design and implement algorithmic solutions to complex problems, particularly under short time constraints
- CLO4 : Apply algorithm design principles to evaluate programs in terms of correctness and running time

Detailed Assessment Description

The final exam will be a long contest (approx 7 problems, 5 hours). Each problem will be worth 100 points, and most or all problems will have one or more subtasks of various point values.

The final exam will be held **in person at CSE labs**. If you are unable to attend in person, please contact me by email in advance so that we can arrange an alternative for you.

If it is necessary to break ties within the same course mark to obtain rankings, e.g. for first place, ties will be broken based on ranks obtained from the final exam according to ICPC scoring rules.

Marks will be scaled up. As a guide, the approximate conversion will be

- 100-150 points 50%
- 200-250 points 65%
- 300-350 points 75%
- 400-500 points 85%

The exact cutoffs will be decided after the exam is completed, taking into account the difficulty of the exam problems.

Assignment submission Turnitin type

Not Applicable

Generative AI Permission Level

Simple Editing Assistance

In completing this assessment, you are permitted to use standard editing and referencing functions in the software you use to complete your assessment. These functions are described below. You must not use any functions that generate or paraphrase passages of text or other media, whether based on your own work or not.
If your Convenor has concerns that your submission contains passages of AI-generated text or media, you may be asked to account for your work. If you are unable to satisfactorily demonstrate your understanding of your submission you may be referred to UNSW Conduct & Integrity Office for investigation for academic misconduct and possible penalties.
For more information on Generative AI and permitted use please see here.

Students may use course resources and prewritten code prepared before the commencement of the contest. Editor and compiler features that do not require internet access or use generative AI are allowed. Language documentation from cppreference.com is allowed.

Problem Sets

Assessment Overview

Students attempt to solve a set of programming problems for each topic, and maintain a diary to reflect on the challenges they encountered.

Submissions are automatically marked against a batch of test cases fitting the specifications provided to students. Students receive feedback of 'Accepted', 'Wrong Answer', 'Time Limit Exceeded', 'Runtime Error' and so on for each attempt, and have unlimited attempts at each problem to get the 'Accepted' verdict. Marks are awarded for the number of problems where the 'Accepted' verdict is received: each set consists of six problems, with (one/two/three/four) solved problem per set to achieve (PS/CR/DN/HD) and the final problems to distinguish top achievers. Students can get help from staff in person during lab time, by email or through the course forum. Students are expected to spend about eight hours per week on these problems.

Course Learning Outcomes

- CLO1 : Apply the C++ programming language to solve programming problems
- CLO2 : Adapt templated data structures and algorithms to meet the requirements of complex problems

- CLO3 : Design and implement algorithmic solutions to complex problems, particularly under short time constraints
- CLO4 : Apply algorithm design principles to evaluate programs in terms of correctness and running time

Detailed Assessment Description

Problem Sets

A problem set will be released on vjudge at the start of each lecture topic (counting the first three lectures as one topic). Students are recommended to complete the problems within about two weeks of the release date, but all problem sets will remain open until the end of the exam period.

Each of the 8 sets will be worth 5% of the overall course mark, for a total of 40%. Marks will be awarded non-linearly based on the number of problems solved:

- 0/5 ↦ 0%
- 1/5 ↦ 2%
- 2/5 ↦ 3%
- 3/5 ↦ 3.75%
- 4/5 ↦ 4.5%
- 5/5 ↦ 5%

Some problem sets will have 6 problems. If you solve all 6 problems, the 6th will be counted towards your lowest other problem set instead. For example "6 5 5 3 4 5 4 5" will be treated as "5 5 5 4 4 5 4 5", with the 'bonus' problem solved in PS1 counting as a fourth solve in PS4.

Solutions must be implemented in C/C++.

Problem Diaries

For each problem set, you must also produce diary entries to

- explain the *process* by which you arrived at your solutions (not the solutions themselves),
- reflect on any challenges encountered and how you overcame them, and
- discuss any collaboration that you participated in with other students.

You are *not* required to write detailed descriptions or proofs for your algorithms as in COMP3121/3821/9101/9801.

Diary entries will be submitted on Formatif. You must make some remarks on every problem up to your target grade (even if only to say that you don't know how to do the question at all). You are encouraged to include any progress made on problems which you did not solve. Any attempt demonstrating a genuine effort to meet the specification listed in this section will be considered satisfactory. If a diary entry is not satisfactory, you can re-attempt it without penalty as many

times as necessary (within reason).

Each problem's diary entry should be a PDF of less than one page, excluding code snippets. A document much shorter than this limit may often be sufficient! You can instead conduct a verbal discussion during lab time, and this will be considered equivalent to a written submission.

The diary entries will collectively be worth 8% of the total course mark.

Students are encouraged to make notes during the problem solving process and submit their diary entries contemporaneously with the problem sets. All diary entries must be submitted before the end of the exam period, and we ask that students do not leave too many diary entries this late, so as not to overload our staff with marking.

Assessment information

There are no deadlines and no late penalties for the problem sets or problem diaries. However, if you have fall several weeks behind on a single problem set, or fall behind on several concurrent problem sets, this can cause a lot of work to be due at once and as a result you might struggle to catch up. In situations where this would be considered, I would generally prefer to exempt you from a problem set so that you can stay up to date with the material being covered in lectures and workshops. Your mark would then be estimated as the minimum of your other problem sets. **Please email me (and cc your lab instructors) if you are concerned about falling behind.**

Assignment submission Turnitin type

Not Applicable

Generative AI Permission Level

Simple Editing Assistance

In completing this assessment, you are permitted to use standard editing and referencing functions in the software you use to complete your assessment. These functions are described below. You must not use any functions that generate or paraphrase passages of text or other media, whether based on your own work or not.
If your Convenor has concerns that your submission contains passages of AI-generated text or media, you may be asked to account for your work. If you are unable to satisfactorily demonstrate your understanding of your submission you may be referred to UNSW Conduct & Integrity Office for investigation for academic misconduct and possible penalties.
For more information on Generative AI and permitted use please see here.

Contests

Assessment Overview

Students will participate in three programming contests during the term.

The first is a long contest of duration two days, while the other two are short contests of duration three hours.

These contests consist of programming problems from recent course material, with immediate feedback for each submission.

Submissions are automatically marked against a batch of test cases fitting the specifications provided to students. Students receive feedback of 'Accepted', 'Wrong Answer', 'Time Limit Exceeded', 'Runtime Error' and so on for each attempt, and have unlimited attempts at each problem to get the 'Accepted' verdict. Marks are again awarded for the number of problems where the 'Accepted' verdict is received, including subtasks.

Course Learning Outcomes

- CLO1 : Apply the C++ programming language to solve programming problems
- CLO2 : Adapt templated data structures and algorithms to meet the requirements of complex problems
- CLO3 : Design and implement algorithmic solutions to complex problems, particularly under short time constraints
- CLO4 : Apply algorithm design principles to evaluate programs in terms of correctness and running time

Detailed Assessment Description

A long contest (5 problems, 48 hours) will be held in week 1. This contest does not test any new material, and is intended to test whether your programming fundamentals are sufficient to proceed to the later stages of the course.

Two short contests (3 problems, 3 hours) will be held in weeks 5 and 9. Each problem will be worth 100 points with a 50 point subtask. Contest 2 will cover Problem-Solving Paradigms, Data Structures I and Dynamic Programming, and contest 3 will cover Graphs, Shortest Paths, Data Structures II and Network Flow.

Each contest will be worth 6% of the overall course mark, for a total of 18%. In contests 2 and 3, marks will be awarded non-linearly, based on the score achieved:

- 0 points 0%
- 50 points 2%
- 100 points 3.5%
- 150 points 4.5%
- 200 points 5%
- 250 points 5.5%
- 300 points 6%

Assessment information

It is unfortunately difficult to schedule these assessments without clashes with your other classes, and without taking up too much of our already limited class time.
I understand that evenings and weekends are not ideal for everyone, so if you are not able to participate during the scheduled times, please contact me and I will be happy to make alternative

arrangements for you.

Assignment submission Turnitin type

Not Applicable

Generative AI Permission Level

Simple Editing Assistance

In completing this assessment, you are permitted to use standard editing and referencing functions in the software you use to complete your assessment. These functions are described below. You must not use any functions that generate or paraphrase passages of text or other media, whether based on your own work or not.
If your Convenor has concerns that your submission contains passages of AI-generated text or media, you may be asked to account for your work. If you are unable to satisfactorily demonstrate your understanding of your submission you may be referred to UNSW Conduct & Integrity Office for investigation for academic misconduct and possible penalties.
For more information on Generative AI and permitted use please see here.

Students may use course resources and prewritten code prepared before the commencement of the contest. Editor and compiler features that do not require internet access or use generative AI are allowed. Language documentation from cppreference.com is allowed.

General Assessment Information

2 bonus marks will be provided for participation or volunteering in the South Pacific ICPC Preliminary Contest on 31st August 2024 or Regional Finals (including mirror contest) on 19th & 20th October 2024. Volunteers' duties include setting up, marking attendance and distributing balloons.

Grading Basis

Standard

Course Schedule

Teaching Week/Module	Activity Type	Content
Week 0 : 2 September - 8 September	Other	ICPC Preliminary Contest (31st August)
Week 1 : 9 September - 15 September	Lecture	Wednesday: Introduction
	Lecture	Thursday: Getting Started
	Workshop	Getting Started
	Laboratory	
	Assessment	Problem Set 1 released
	Assessment	Contest 1
Week 2 : 16 September - 22 September	Lecture	Wednesday: Problem-Solving Paradigms
	Lecture	Thursday: Data Structures
	Workshop	Problem-Solving Paradigms
	Laboratory	
	Assessment	Problem Set 2 released
Week 3 : 23 September - 29 September	Lecture	Wednesday: Data Structures (continued)
	Lecture	Thursday: Dynamic Programming
	Workshop	Data Structures
	Laboratory	
	Assessment	Problem Set 3 released
Week 4 : 30 September - 6 October	Lecture	Wednesday: Dynamic Programming (continued)
	Lecture	Thursday: Graphs
	Workshop	Dynamic Programming
	Laboratory	
	Assessment	Problem Set 4 released
Week 5 : 7 October - 13 October	Lecture	Wednesday: Graphs (continued)
	Lecture	Thursday: Shortest Paths
	Workshop	Graphs
	Laboratory	
	Assessment	Problem Set 5 released
	Assessment	Contest 2
Week 6 : 14 October - 20 October	Lecture	Wednesday: TBA
	Lecture	Thursday: TBA
Week 7 : 21 October - 27 October	Lecture	Wednesday: Shortest Paths (continued)
	Lecture	Thursday: Flow Networks
	Workshop	Shortest Paths
	Laboratory	
	Assessment	Problem Set 6 released
Week 8 : 28 October - 3 November	Lecture	Wednesday: Flow Networks (continued)
	Lecture	Thursday: Mathematics
	Workshop	Flow Networks
	Laboratory	
	Assessment	Problem Set 7 released
Week 9 : 4 November - 10 November	Lecture	Wednesday: Mathematics (continued)
	Lecture	Thursday: Computational Geometry
	Workshop	Mathematics
	Laboratory	
	Assessment	Problem Set 8 released
	Assessment	Contest 3
Week 10 : 11 November - 17 November	Lecture	Wednesday: Mathematics (continued)
	Lecture	Thursday: Exam Revision

	Workshop	Computational Geometry
	Laboratory	

Attendance Requirements

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

General Schedule Information

The full schedule will be published on Moodle, where it will be conveniently formatted in a table.

Course Resources

Prescribed Resources

The course website will contain all relevant resources, and there are no textbooks required for the course.

Recommended Resources

The following textbooks are suitable for reference:

- Any algorithms textbook, such as *Introduction to Algorithms* (Cormen, Leiserson, Rivest and Stein) or *Algorithm Design* (Kleinberg and Tardos)
- *Programming Challenges* (Skiena and Revilla)
- *Competitive Programming 4* (Halim). A free copy of Competitive Programming 1 can be found at CPBook.

Students may find the following online resources helpful:

- CP-Algorithms (and the more comprehensive but Russian Emaxx)
- Topcoder Tutorials
- There are many helpful blogs, particularly on Codeforces (a partial list here). Some select choices:
  - Range Trees
  - Dynamic Programming On Trees
  - Geometry
  - Dynamic Programming Optimizations

Further practice problems can be found in many places, such as:

- Codeforces
- AtCoder
- topcoder
- codechef
- USACO

Students are encouraged to ask course staff for further resources.

Course Evaluation and Development

This course is evaluated each session using the myExperience system.

In the previous offering of this course, students reported that diaries were sometimes too much of a chore.

Based on their comments, we will accept verbal discussions in place of a written diary, and not expect all students to write about all problems.

Staff Details

Position	Name	Email	Location	Phone	Availability	Equitable Learning Services Contact	Primary Contact
Convenor	Raveen De Silva		K17 202			Yes	No
	COURSE EMAIL					Yes	Yes

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.