UNSW Course Outline

COMP1511 Programming Fundamentals - 2024

Published on the 25 Aug 2024

General Course Information

Course Code : COMP1511
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 : 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

From recent innovations in AI like self-driving cars to humanoid robotics navigating complex environments, leapfrogs in battery technology to sequencing the human genome - the world is benefiting and evolving thanks to computer systems. At the core of all these systems are computers executing instructions to solve exciting problems.

In this course, you will learn the fundamentals of how we instruct computers to solve problems. You will explore the architecture and mechanics of how computers operate and how you can translate real-world problems to computer programs that solve these problems.

The concepts you learn will provide a foundation for your future endeavours in computing and, we hope, will begin to change the way you think about real-world problems.

This course is an introductory course to the basics of computer programming and Computer Science. It is intended as an introduction to studying further in Computer Science or related fields. Topics include:

- Fundamental programming concepts
- Introduction to Computer Science
- The C programming language and use of a C compiler
- Programming style
- Program design and organisation concepts
- Program testing and debugging

Course Aims

The importance of this course lies in its role as the foundation of your programming journey, providing essential knowledge and skills vital for your success in the field. By focusing on proficiency in the high-level programming language C and fostering problem-solving abilities, this course equips you with the fundamental tools and mindset necessary to think like a programmer.

As the first course in the program, it plays a crucial role in setting the stage for your future learning. It serves as a prerequisite for many of the core courses, ensuring that all students begin with a solid understanding of the fundamental concepts required to progress further. By establishing a common knowledge base and skills, this course ensures that everyone starts on an equal footing and can effectively tackle more advanced topics.

This course intends to guide you through the initial stages of your programming education, imparting technical proficiency in C and the ability to approach problems systematically and think critically. By emphasizing problem-solving strategies, debugging techniques, and testing methodologies, the course aims to instill in you a resilient and adaptable mindset that will serve as a solid foundation for your future development as a programmer.

Course Learning Outcomes

Course Learning Outcomes
CLO1 : Apply C programming language to solve simple decision, looping, array, and linked list problems programmatically
CLO2 : Review the produced code against specification criteria by applying testing techniques
CLO3 : Apply basic data structures, such as arrays and linked lists, to solve complex problems
CLO4 : Read and understand coding solutions.

Course Learning Outcomes	Assessment Item
CLO1 : Apply C programming language to solve simple decision, looping, array, and linked list problems programmatically	• Problem Sets • Assignment 1 • Assignment 2 • Final Exam
CLO2 : Review the produced code against specification criteria by applying testing techniques	• Problem Sets • Assignment 1 • Assignment 2 • Final Exam
CLO3 : Apply basic data structures, such as arrays and linked lists, to solve complex problems	• Problem Sets • Assignment 1 • Assignment 2 • Final Exam
CLO4 : Read and understand coding solutions.	• Problem Sets • Assignment 1 • Assignment 2 • Final Exam

Learning and Teaching Technologies

EdStem | Custom LMS | YouTube | Microsoft Teams

Assessments

Assessment Structure

Assessment Item	Weight	Relevant Dates
Problem Sets Assessment Format: Individual	15%	Start Date: Weekly Due Date: Weekly
Assignment 1 Assessment Format: Individual	20%	Start Date: Week 4 Due Date: Week 7
Assignment 2 Assessment Format: Individual	25%	Start Date: Week 7 Due Date: Week 10
Final Exam Assessment Format: Individual	40%	Due Date: during Exam Period

Assessment Details

Problem Sets

Assessment Overview

Each problem set will be submitted using the "give" system. All students will need to submit solutions to the Problem Sets each week (weeks 2-10).

Problem sets will be marked automatically one week after the due date. When marking is complete you can see marks online using the course website.

Problem Sets have an indicator to help you choose which problems to do -- you should prioritise the one-dot exercises first, then two-dot, then three. You must complete all one-dot and two-dot exercises to get full marks in the problem set for the Week. Any three-dot exercises you complete will form extra bonus marks to compensate for any other missing marks in the problem set component.

There are no marks for Problem Set 1, it's there to help you get started. You will see a 0 as a total for Problem Set 1, as it is not counted towards your final mark.

Problem sets are capped at 15 ***marks*** (there are 4 possible bonus marks from the three-dot exercises that can bring you up to a total of 15 if you missed out on any other marks in the one- or two-dot exercises). ***Completing just the one- and two-dot exercises every week*** can give you the full 15 marks needed in this component.

Course Learning Outcomes

- CLO1 : Apply C programming language to solve simple decision, looping, array, and linked list

problems programmatically
- CLO2 : Review the produced code against specification criteria by applying testing techniques
- CLO3 : Apply basic data structures, such as arrays and linked lists, to solve complex problems
- CLO4 : Read and understand coding solutions.

<u>**Assignment submission Turnitin type**</u>

Not Applicable

<u>**Generative AI Permission Level**</u>

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.

COMP1511 Specific Information

You are permitted to use the tools **dcc-help** and **dcc-sidekick** to help you understand the error messages you may get when compiling the code you have written.

You are permitted to use **autotest-help** to help you understand why your code may not be passing the automated tests.

You are not permitted to submit code generated by automatic AI tools such as Github Copilot, ChatGPT, Google Bard in COMP1511 for lab exercises. Submitting code generated by Github Copilot, ChatGPT, Google Bard and similar tools will be treated as plagiarism.

Our reasoning behind our decisions:

Systems such as Github Copilot and ChatGPT based on large language models or other generative artificial intelligence techniques, look likely to become heavily used by programmers. However, you need a good understanding of the language you are coding in and the systems involved before you can effectively use these tools. Using these tools to generate code for COMP1511 instead of writing the code yourself will hinder your learning.

Assignment 1

Assessment Overview

There are two assessable programming assignments. Assignments allow you to practice what you have learned on relatively large problems (compared to the small exercises in the labs). Assignments are a very important part of this course, therefore it is essential that you attempt them yourself. Collaboration with other students is limited to discussion of fundamentals, not any discussion of assignment specifics.

- Assignment 1, 20%

Assignment 1 is focused on the topic of arrays.

Assignments will be automarked, and feedback on programming style will be given by tutors.

Course Learning Outcomes

- CLO1 : Apply C programming language to solve simple decision, looping, array, and linked list problems programmatically
- CLO2 : Review the produced code against specification criteria by applying testing techniques
- CLO3 : Apply basic data structures, such as arrays and linked lists, to solve complex problems
- CLO4 : Read and understand coding solutions.

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.

COMP1511 Specific Information

You are permitted to use the tools **dcc-help** and **dcc-sidekick** to help you understand the error messages you may get when compiling the code you have written.

You are permitted to use **autotest-help** to help you understand why your code may not be passing the automated tests.

You are not permitted to submit code generated by automatic AI tools such as Github Copilot, ChatGPT, Google Bard in COMP1511 for assignments. Submitting code generated by Github Copilot, ChatGPT, Google Bard and similar tools will be treated as plagiarism.

Our reasoning behind our decisions:

Systems such as Github Copilot and ChatGPT based on large language models or other generative artificial intelligence techniques, look likely to become heavily used by programmers. However, you need a good understanding of the language you are coding in and the systems involved before you can effectively use these tools. Using these tools to generate code for COMP1511 instead of writing the code yourself will hinder your learning.

Assignment 2

Assessment Overview

There are two assessable programming assignments. Assignments allow you to practice what you have learned on relatively large problems (compared to the small exercises in the labs). Assignments are a very important part of this course, therefore it is essential that you attempt them yourself. Collaboration with other students is limited to discussion of fundamentals, not any discussion of assignment specifics.

- Assignment 2, 25%

Assignment 2 is a relatively large problem, focusing on linked lists.

Assignments will be automarked, and feedback on programming style will be given by tutors.

Course Learning Outcomes

- CLO1 : Apply C programming language to solve simple decision, looping, array, and linked list problems programmatically
- CLO2 : Review the produced code against specification criteria by applying testing techniques
- CLO3 : Apply basic data structures, such as arrays and linked lists, to solve complex problems
- CLO4 : Read and understand coding solutions.

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.

COMP1511 Specific Information

You are permitted to use the tools **dcc-help** and **dcc-sidekick** to help you understand the error messages you may get when compiling the code you have written.

You are permitted to use **autotest-help** to help you understand why your code may not be passing the automated tests.

You are not permitted to submit code generated by automatic AI tools such as Github Copilot, ChatGPT, Google Bard in COMP1511 for assignments. Submitting code generated by Github Copilot, ChatGPT, Google Bard and similar tools will be treated as plagiarism.

Our reasoning behind our decisions:

Systems such as Github Copilot and ChatGPT based on large language models or other generative artificial intelligence techniques, look likely to become heavily used by programmers. However, you need a good understanding of the language you are coding in and the systems involved before you can effectively use these tools. Using these tools to generate code for COMP1511 instead of writing the code yourself will hinder your learning.

Final Exam

Assessment Overview

3-hour-long in-person exam taking place in exam period.

The exam will contain implementation tasks that will require you to write C programs. It will also contain sections that require you to read code or answer questions to show your knowledge of programming.

During this exam, you will be able to execute, debug and test your answers. The implementation tasks will be similar to those encountered in your weekly Problem Sets.

We will provide you with sample questions in the last week of the course.

Special Exam Requirements

There are two requirements for the final exam.

Requirement#1: on the final exam you must solve a task by writing a program that uses an array. There will be multiple, clearly marked, questions that will involve the use of an array. You must pass one of these questions (by receiving at least 50% of the available marks) to meet this requirement.

Requirement#2: on the final exam you must solve a task by writing a program that uses a linked list. There will be multiple, clearly marked, questions that will involve using a linked list. You must pass one of these questions (by receiving at least 50% of the available marks) to meet this requirement.

You cannot pass COMP1511 unless you achieve both the above requirements.

Course Learning Outcomes

- CLO1 : Apply C programming language to solve simple decision, looping, array, and linked list problems programmatically
- CLO2 : Review the produced code against specification criteria by applying testing techniques
- CLO3 : Apply basic data structures, such as arrays and linked lists, to solve complex problems
- CLO4 : Read and understand coding solutions.

Assignment submission Turnitin type

Not Applicable

Hurdle rules

There are two requirements for the final exam.

Requirement 1: on the final exam, you must solve a task by writing a program that uses an array. There will be multiple, clearly marked questions that will involve the use of an array. You must pass one of these questions (by receiving at least 50% of the available marks) to meet this requirement.

Requirement 2: on the final exam, you must solve a task by writing a program that uses a linked list. There will be multiple, clearly marked questions that will involve using a linked list. You must pass one of these questions (by receiving at least 50% of the available marks) to meet this requirement.

You cannot pass COMP1511 unless you achieve both the above requirements.

Generative AI Permission Level

No Assistance

This assessment is designed for you to complete without the use of any generative AI. You are not permitted to use any generative AI tools, software or service to search for or generate information or answers.
For more information on Generative AI and permitted use please see here.

COMP1511 Specific Information

You will not have access to any generative AI tools during the exam.

You will not have access to **dcc-help**, **dcc-sidekick** or **autotest-help** during the exam.

General Assessment Information

Grading Basis

Standard

Requirements to pass course

In addition to scoring 50 or above overall in the course, COMP1511 has two requirements for the final exam:

Requirement 1: on the final exam, you must solve a task by writing a program that uses an array. There will be multiple, clearly marked questions that will involve the use of an array. You must pass one of these questions (by receiving at least 50% of the available marks) to meet this requirement.

Requirement 2: on the final exam, you must solve a task by writing a program that uses a linked list. There will be multiple, clearly marked questions that will involve using a linked list. You must pass one of these questions (by receiving at least 50% of the available marks) to meet this requirement.

You can not pass COMP1511 unless you achieve both the above requirements AND score an overall course grade of 50 and above.

Course Schedule

Teaching Week/Module	Activity Type	Content
Week 1 : 9 September - 15 September	Lecture	Course intro UNIX + Tools What is a program
	Lecture	Variables/Constants
Week 2 : 16 September - 22 September	Lecture	Control Flow
	Lecture	Custom Data Types
Week 3 : 23 September - 29 September	Lecture	Procedures and Functions
	Lecture	Static Arrays
Week 4 : 30 September - 6 October	Lecture	2D Arrays
	Lecture	Strings
Week 5 : 7 October - 13 October	Lecture	No Lecture - PUBLIC HOLIDAY (A recording will be provided) Lecture Program 1: Arrays
	Lecture	Pointers
Week 7 : 21 October - 27 October	Lecture	Dynamic arrays Memory
	Lecture	Memory (heap vs stack) Basic linked list
Week 8 : 28 October - 3 November	Lecture	Linked Lists
	Lecture	Lecture: Linked Lists
Week 9 : 4 November - 10 November	Lecture	Lecture Program 2: Linked Lists
	Lecture	Extra Content (Non-Examinable)
Week 10 : 11 November - 17 November	Lecture	Exam details
	Lecture	Revision of course content

Attendance Requirements

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

Course Resources

Recommended Resources

This book is an optional text for the course, you are NOT required to have it. You can find it at UNSW Library (https://primoa.library.unsw.edu.au/permalink/61UNSW_INST/1m02euc/

alma9950458840001731) or at the UNSW Bookshop (Print: https://www.bookshop.unsw.edu.au/details.cgi?ITEMNO=9781486010974 ; Digital: https://unswbookshop.vitalsource.com/products/-v9781486010981)

Course Evaluation and Development

At the end of every term, COMP1511 students are invited to provide their feedback about the course through the UNSW myExperience online survey system. This is used to assess the quality of the course so that we can make ongoing improvements. We do take this feedback seriously and use it to improve the course materials and their delivery. Students are also encouraged to provide informal feedback during the term by letting the Lecturer in Charge or any of the course staff, know of any problems, as soon as they arise. Suggestions will be listened to openly, positively, constructively, and thankfully, and every reasonable effort will be made to address them. Recent myExperience evaluations showed that students were highly satisfied with most aspects of the course. However, there are always things that can be improved, some changes that we are making this term:

- Introduction of more pair programming exercises every week to encourage a more collaborative environment and to provide a supportive environment in which you can learn to code.
- Introduce more debugging-style questions to give you ways in which to practice learning how to code.
- Improving the Formatif module to provide students with an avenue to get feedback on their coding style and efficiency and to help them get better at solving problems.
- Provide more short videos to cover some of the key course concepts that can be used for revision.
- Targetted revision sessions every fortnight based on your needs!

CSE may also run its own survey, midway through the term, to elicit feedback while courses are still running. This course improves only because we see the difficulties that students have and try to adjust things so that you get to learn what you need. If anything's not working for you, please let us know and we'll do whatever we can to help and hopefully help students in later cohorts as well.

Staff Details

Position	Name	Email	Location	Phone	Availability	Equitable Learning Services Contact	Primary Contact
Lecturer	Angela Finlayson					Yes	No
Administrator	Tammy Zhong					No	No
	Nicole Luong					No	No
	Sofia De Bellis					No	No
	Daniel Slachov					No	No
	COURSE EMAIL					No	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.