



UNSW

UNSW Course Outline

COMP9164 Concepts of Programming Languages - 2024

Published on the 27 Aug 2024

General Course Information

Course Code : COMP9164

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

Units of Credit : 6

Useful Links

[Handbook Class Timetable](#)

Course Details & Outcomes

Course Description

This course discusses and relates a range of programming language concepts and paradigms, including imperative, object-oriented, functional, concurrent & parallel programming. It covers the theoretical foundations (syntax, operational, axiomatic and denotational semantics) as well as

practical development and implementation aspects (dynamic and strong typing, polymorphism, overloading, automatic memory management, and concurrency). Due to the course topics, a variety of programming languages will be studied and used. Assignments will be in the programming language Haskell, but no previous knowledge of Haskell is assumed.

Course Aims

This course aims to impart the core concepts and principles of programming language design. The course will improve students' appreciation of why programming languages are the way they are, how programming language features can be designed to guarantee desirable properties, and how programming languages are defined mathematically and implemented practically. The practical and theoretical skills in this course can deepen your understanding of what you're doing as a programmer, and make current trends in programming language design intelligible; it also provides essential skills for anyone looking to develop their own programming language, or get into programming language research.

Relationship to Other Courses

This course is equivalent to and taught in combination with COMP3161.

Course Learning Outcomes

Course Learning Outcomes
CLO1 : Distinguish superficial (i.e., syntactic) from fundamental (i.e., semantic) differences of programming languages
CLO2 : Read, write and reason about programs in various programming language paradigms, including imperative, object-oriented, functional and concurrent programming
CLO3 : Use the basic tools of language design and analysis, including formal languages and formal descriptions of static and dynamic semantics
CLO4 : Use modern language features, such as dynamic and strong typing, polymorphism, overloading, automatic memory management, and concurrency features

Course Learning Outcomes	Assessment Item
CLO1 : Distinguish superficial (i.e., syntactic) from fundamental (i.e., semantic) differences of programming languages	<ul style="list-style-type: none">• Programming assignments• Theory Assignment
CLO2 : Read, write and reason about programs in various programming language paradigms, including imperative, object-oriented, functional and concurrent programming	<ul style="list-style-type: none">• Programming assignments
CLO3 : Use the basic tools of language design and analysis, including formal languages and formal descriptions of static and dynamic semantics	<ul style="list-style-type: none">• Theory Assignment• Programming assignments
CLO4 : Use modern language features, such as dynamic and strong typing, polymorphism, overloading, automatic memory management, and concurrency features	<ul style="list-style-type: none">• Programming assignments

Learning and Teaching Technologies

Moodle - Learning Management System | Zoom | Echo 360 | EdStem | CSE Systems

Assessments

Assessment Structure

Assessment Item	Weight	Relevant Dates
Final Exam Assessment Format: Individual	50%	
Programming assignments Assessment Format: Individual	35%	
Theory Assignment Assessment Format: Individual	15%	

Assessment Details

Final Exam

Assessment Overview

The final exam will focus on theoretical knowledge. It will be assessed against marking criteria. Estimated effort 2-3 hours.

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](#).

You are *not* allowed to use AI tools (such as ChatGPT or GitHub Copilot) to help you with technical content, or to develop definitions and proofs.

You are allowed to use AI tools for non-technical purposes; for example, to help polish your grammar in essay questions.

Programming assignments

Assessment Overview

There will be two programming assignments involving the implementation of a programming language.

All assignments are individual; i.e., no team work of any kind is permitted. The assignments are an extremely important part of the course. They are an essential way of learning the practical skills you need to acquire. Any plagiarism in assignments may lead to serious consequences.

For assignments 1 and 2, you will have approximately two weeks from release of the specification until the submission deadline.

Indicative effort: write a small-medium scale Haskell program. Assignment will be marked with a mix of automarking and manual assessment against marking criteria. Automarking results will be available online.

Course Learning Outcomes

- CLO1 : Distinguish superficial (i.e., syntactic) from fundamental (i.e., semantic) differences of programming languages
- CLO2 : Read, write and reason about programs in various programming language paradigms, including imperative, object-oriented, functional and concurrent programming
- CLO3 : Use the basic tools of language design and analysis, including formal languages and formal descriptions of static and dynamic semantics
- CLO4 : Use modern language features, such as dynamic and strong typing, polymorphism, overloading, automatic memory management, and concurrency features

Assignment submission Turnitin type

This is not a Turnitin assignment

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](#).

You are *not* allowed to use AI tools (such as ChatGPT or GitHub Copilot) to help you with technical content, or to develop definitions and proofs.

You are allowed to use AI tools for non-technical purposes; for example, to help polish your grammar in essay questions.

Theory Assignment

Assessment Overview

There will be a smaller assignment that focuses on theoretical topics and proofs. You will be given a series of proof exercises to complete and submit. Some of the questions given will be similar to exam questions.

All assignments are individual; i.e., no team work of any kind is permitted. The assignments are an extremely important part of the course. They are an essential way of learning the practical skills you need to acquire. Any plagiarism in assignments may lead to serious consequences.

For assignment 0, you will have approximately one week to complete it.

Indicative effort: write up a few pages of mathematics. Assignment will be marked against assessment criteria, and individual feedback will be provided online.

Course Learning Outcomes

- CLO1 : Distinguish superficial (i.e., syntactic) from fundamental (i.e., semantic) differences of programming languages
- CLO3 : Use the basic tools of language design and analysis, including formal languages and formal descriptions of static and dynamic semantics

Assignment submission Turnitin type

This is not a Turnitin assignment

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](#).

You are *not* allowed to use AI tools (such as ChatGPT or GitHub Copilot) to help you with technical content, or to develop definitions and proofs.

You are allowed to use AI tools for non-technical purposes; for example, to help polish your grammar in essay questions.

General Assessment Information

Grading Basis

Standard

Course Schedule

Teaching Week/Module	Activity Type	Content
Week 0 : 2 September - 8 September	Lecture	Nothing
Week 1 : 9 September - 15 September	Lecture	Intro, Haskell, Induction
Week 2 : 16 September - 22 September	Lecture	Natural Deduction, Rule Induction, Syntax
Week 3 : 23 September - 29 September	Lecture	Assignment 0, Operational Semantics, λ Calculus
Week 4 : 30 September - 6 October	Lecture	Imperative Programming, MinHS
Week 5 : 7 October - 13 October	Lecture	Assignment 1, Abstract Machines, Environments
Week 6 : 14 October - 20 October	Lecture	Flexibility Week
Week 7 : 21 October - 27 October	Lecture	Properties, Type safety, Type theory
Week 8 : 28 October - 3 November	Lecture	Polymorphism, Type Inference, Assignment 2
Week 9 : 4 November - 10 November	Lecture	Abstract Data Types, Overloading and Subtyping
Week 10 : 11 November - 17 November	Lecture	Concurrency, Revision

Attendance Requirements

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

General Schedule Information

The schedule is tentative and may evolve as the course proceeds.

Further information is on the course website at CSE <https://www.cse.unsw.edu.au/~cs3161> .

Staff Details

Position	Name	Email	Location	Phone	Availability	Equitable Learning Services Contact	Primary Contact
COURSE ACCOUNTANT	Thomas Sewell					Yes	No
	Miki Tanaka					No	No
	Robert Sison					No	No
	COURSE ACCOUNTANT					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 policies. 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 Plagiarism

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.