UNSW Course Outline

COMP1010 The Art of Computing - 2024

Published on the 01 Feb 2024

General Course Information

Course Code : COMP1010
Year : 2024
Term : Term 1
Teaching Period : T1
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 : Undergraduate
Units of Credit : 6

Useful Links

Handbook Class Timetable

Course Details & Outcomes

Course Description

This course aims to provide a grounding in computational thinking for anyone who wants one. It assumes no previous programming background, but does assume that all incoming students have used digital devices, such as tablets and smart phones, for a range of tasks (e.g. social

networking, reading, essay writing, etc.). The course will use Python as the programming medium and use real world examples from a variety of domains to motivate understanding.

Course Aims

This course aims to provide a grounding in programming and computational thinking for any undergraduate interested in learning these essential skills. It assumes no previous programming background, but does assume that all incoming students have used digital devices, such as tablets and smart phones, for a range of tasks (e.g. social networking, reading, essay writing, etc.). The course will use Python as the programming medium.

The course is a purely elective course, intended to be available as a General Education course for non-CSE students.

Topics: computer systems, computational problem solving, spreadsheets, programming in Python.

Relationship to Other Courses

COMP1010 is intended to be an introduction to computer programming designed for students who have nobackground in computer programming and who are studying a degree outside of Computer Science and Engineering.

It should **not** be taken by students who have studied or are currently enrolled in ENGG1811, COMP1511, or any other COMP course.

Course Learning Outcomes

Course Learning Outcomes
CLO1 : Use a spreadsheet for simple data management tasks.
CLO2 : Write Python programs to solve simple computational problems.
CLO3 : Solve problems via computer systems.
CLO4 : Build simple human-centered interfaces to computers.

Course Learning Outcomes	Assessment Item
CLO1 : Use a spreadsheet for simple data management tasks.	• Lab Exercises • Final Exam
CLO2 : Write Python programs to solve simple computational problems.	• Project - Proposal, Implementation, Presentation and Feedback • Lab Exercises • Final Exam
CLO3 : Solve problems via computer systems.	• Project - Proposal, Implementation, Presentation and Feedback • Lab Exercises • Final Exam
CLO4 : Build simple human-centered interfaces to computers.	• Project - Proposal, Implementation, Presentation and Feedback • Lab Exercises • Final Exam

Learning and Teaching Technologies

Zoom | EdStem | Echo 360

Learning and Teaching in this course

Lectures will introduce concepts and show examples.

Tutorials will reinforce concepts and provide additional examples.

Lab work provides essential practice in programming.

Project allows students to solve a significant prolem which matters to them.

In order to succeed in this course, students are required to consistently apply the concepts and skills taught in lectures and tutorials. It is highly recommended (if not essential) that students do practice questions outside of those being assessed in order to develop the required skills.

Keeping up with the content as it is taught is important as each week builds on the the content

from previous weeks. By week 4, it is expected that students are confident in the content taught in weeks 1-3 and are able to apply the material in a more complex environment.

Assessments

Assessment Structure

Assessment Item	Weight	Relevant Dates
Lab Exercises Assessment Format: Individual	20%	Due Date: Week 1: 12 February - 18 February, Week 2: 19 February - 25 February, Week 3: 26 February - 03 March, Week 4: 04 March - 10 March, Week 5: 11 March - 17 March, Week 7: 25 March - 31 March, Week 8: 01 April - 07 April, Week 9: 08 April - 14 April, Week 10: 15 April - 21 April
Project - Proposal, Implementation, Presentation and Feedback Assessment Format: Individual	35%	Due Date: Week 5: 11 March - 17 March, Week 10: 15 April - 21 April
Final Exam Assessment Format: Individual	45%	

Assessment Details

Lab Exercises

Assessment Overview

In weeks 1-5 and 7-10 you will need to complete a series of lab exercises. These will typically be programming exercises asking you to solve a particular problem in python. To receive marks for these labs you will need to get them marked by your lab coordinator in your allocated lab time in the week after the lab is released. For example, you will need to get the week 1 lab exercises marked in your week 2 lab. If you finish your exercises early, you can, of course, get them marked in the week they were assigned.

The lab coordinator will provide both marks and feedback on the lab exercises.

Note: It is likely that the total number of marks for labs for the term will be somewhere between 20 and 30. This will only be confirmed in week 10. You can expect each lab mark to be worth between 0.66% of your course grade and 1% of your course grade.

Course Learning Outcomes

- CLO1 : Use a spreadsheet for simple data management tasks.
- CLO2 : Write Python programs to solve simple computational problems.

- CLO3 : Solve problems via computer systems.
- CLO4 : Build simple human-centered interfaces to computers.

Assignment submission Turnitin type

This is not a Turnitin assignment

Project - Proposal, Implementation, Presentation and Feedback

Assessment Overview

In the project, you will implement an application in Python solving a problem of your choice. You will be assessed on how well you were able to write the code for this app as well as the design of its interface. While available time and skills are limitations that will need to be taken into account, the problem you choose to solve is up to you. Your project idea will need to be approved by your tutor. You are encouraged to work with a partner on your project, especially if you're new to programming, but you are not required to and can complete the project on your own if you wish.

Your tutor will assess your work, awarding marks according to criteria in the project specification, and provide feedback.

Weighting:

Proposal 10%
Implementation 20%
Demo 0%*
Feedback 5%

* While the project demonstration is not assessed in itself, failure to attend, demonstrate and respond to questions about one's project may impact the mark awarded for the implementation.

Course Learning Outcomes

- CLO2 : Write Python programs to solve simple computational problems.
- CLO3 : Solve problems via computer systems.
- CLO4 : Build simple human-centered interfaces to computers.

Assignment submission Turnitin type

This is not a Turnitin assignment

Final Exam

Assessment Overview

The final exam will be held in person on the CSE lab machines in the UNSW exam period. You will have 3 hours to complete it. It will involve programming exercises, but may also include multiple choice and short answer.

The exam is largely auto-marked, and the mark is subject to scaling.

Course Learning Outcomes

- CLO1 : Use a spreadsheet for simple data management tasks.
- CLO2 : Write Python programs to solve simple computational problems.
- CLO3 : Solve problems via computer systems.
- CLO4 : Build simple human-centered interfaces to computers.

Hurdle rules

Students are required to achieve a grade of at least 50% on the final exam in order to pass the course. Students who do not meet this requirement, but otherwise achieve reasonable marks in the course, may be offerred a second chance to complete the competency requirement for the course. If this is achieved, the hurdle requirement attached to the final exam will be waived.

General Assessment Information

Grading Basis

Standard

Requirements to pass course

There is a competency requirement for COMP1010. You must score 50% in the final exam to pass this course. Students with good attempts at other coursework who do not achieve competency on the final exam may (at the discretion of the LIC) be offered an additional opportunity to demonstrate competency.

Course Schedule

Teaching Week/Module	Activity Type	Content
Week 1 : 12 February - 18 February	Lecture	Course Introduction Python and Programming Fundamentals
	Tut-Lab	Python and Programming Fundamentals
Week 2 : 19 February - 25 February	Lecture	Python and Programming Fundamentals
	Tut-Lab	Python and Programming Fundamentals
Week 3 : 26 February - 3 March	Lecture	Python and Programming Fundamentals Web Development Introduction
	Tut-Lab	Python and Programming Fundamentals
Week 4 : 4 March - 10 March	Lecture	Web Application Development
	Tut-Lab	Web Application Development Project Approval
Week 5 : 11 March - 17 March	Lecture	Web Application Development
	Tut-Lab	Web Application Development
	Assessment	Project Proposal Due
Week 6 : 18 March - 24 March	Other	Flex Week: No classes.
Week 7 : 25 March - 31 March	Lecture	Web Application Development
	Tut-Lab	Web Application Development
Week 8 : 1 April - 7 April	Lecture	Web Application Development
	Tut-Lab	Web Application Development
Week 9 : 8 April - 14 April	Lecture	Web Application Development
	Tut-Lab	Web Application Development
Week 10 : 15 April - 21 April	Lecture	Web Application Development
	Tut-Lab	Web Application Development
	Assessment	Project Implementation Due
	Assessment	Project Demonstration
	Assessment	Project Feedback Due

Attendance Requirements

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

General Schedule Information

Weeks 1-5, 7-10: 4 hours of lecture, 1 hour of tutorial, 2 hours of lab.

Lab work due each week.

Project milestones are listed below.

Course Resources

Prescribed Resources

None.

Recommended Resources

There are no formal textbooks for this course, but students may find the following FREE book helpful for learning and practicing python programming

- *Think Python 2nd Edition* by Allen B. Downey, available at https://greenteapress.com/wp/think-python-2e/

Other FREE resources which have been discovered and recommended by past students of this course include:

- An app (free version available on Android and iOS): https://getmimo.com/
- An interactive website: https://www.w3schools.com/python/
- Tutorial-style articles with accompanying exercises: https://pythonbasics.org/

Additional Costs

None.

Course Evaluation and Development

This course is evaluated each session using the myExperience system as well as feedback from students and tutors throughout each term.

- **Lab exercises matching lecture content more closely (23T3)**
  In introducing more lab material in 23T1, the lab work diverged from the content being taught in lectures. Students responded to this negatively so we are putting effort in to re-align the content in the labs to what is being taught in lectures.
- **More modular labs (23T3)**
  Due to concerns that the lab books make it hard to keep track of what a student has completed and has yet to complete, we are trialling a system whereby exercises are released to students individually and submitted individually and students can check which submissions they have completed and which the have not through the View Submissions page.
- **Maintenance of Current Practices (22T3)**
  Feedback from tutors, and students when asked, reflect that the demonstrations and lab books (introduced in 21T3 and 22T1 respectively) are fulfilling their intended goals. The insertion of a specific lecture outlining the purpose and use of consultations resulted in a reduction (from 3 in 22T1 to 0 in 22T2) of students reaching the end of the course without completing at least a minimal project.
- **More Practice and More Gradual Learning Curve (22T3)**
  To continue to address feedback about the pace of the course being too fast, while wanting to keep the amount of content, we will be adding more resources to aid students in doing more practice and building familiarity with concepts. We will also consider removing the spreadsheets topic for future terms. Additional consultations will also be added in the early

weeks of the course to assist in the development of fundamental programming and Python skills.

- More Student Support (22T2)
  In 2022 T1 some students indicated that although consultations were available, they were reluctant to use them because they didn't know what to expect in them. In response to this we will include specific, scheduled, targeted consultations throughout the term to support students with common problems. This includes helping students set up their computers (around week 4 and week 5) and helping students overcome obstacles they're facing in developing their project (weeks 7,8,9 and 10).
- Slower Pace and More Tutorial and Lab Questions (21T3)
  Students in 2021 T2 indicated that in some areas the pace of the course was too fast. In response to this we have:
  - Reduced the number of topics (removing databases, recursion) (in order to spend more time on introducing earlier topics more slowly).
  - Created more tutorial and lab questions on certain topics to help students gain more competence in key areas.
- Introduction of Project Demonstrations (21T3)
  In 2021 T3 we trialled the process of students demonstrating their projects as part of the assessment process. Feedback from both students and staff deemed this a success and so will continue into 22T1.
- Introduction of Lab Books (22T1)

  In 2021 T3, after increasing the number of lab questions, they were disproportionately distributed over the labs (particularly at the end of the Python and Programming Fundamentals topic). To improve the distribution of work, the following changes will be made:
  - At the start of each core topic (spreadsheets, Python and programming fundamentals, web apps) a lab book containing all the lab work for that topic will be released. This is intended to increase the transparency about the workload involved in each topic and allow students more control over the distribution of their workload.
  - Due dates will be assigned to lab questions at the end of each lecture. This is intended to communicate to the students at what point in the content they should be capable of completing certain questions in the lab book.

  This combination we hope will allow students to do their lab work as early as possible (and thereby avoid "bunching up" of topics at the end of each topic) while only attempting work which has already been taught in lectures.
- Structure of Examples Provided in Lectures (Ongoing)
  Mixed feedback was provided throughout 21T2 and 21T3 in response to the structure of examples provided in lectures. Some students prefer a single file with small increments made to develop a single comprehensive sample code. Other students prefer many small isolated examples of each item being taught. We will continue to seek feedback throughout the term and try to cater to as many students as possible, within the current cohort, at the same time.

Students are also encouraged to provide informal feedback during the session, and to let 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.

Staff Details

Position	Name	Email	Location	Phone	Availability	Equitable Learning Services Contact	Primary Contact
Lecturer	Sim Mautner				E-mail to arrange a consultation time.	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 course 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)