UNSW Course Outline

COMP9021 Principles of Programming - 2024

Published on the 29 Jan 2024
Course Code : COMP9021
Year : 2024
Term : Term 1
Teaching Period : T1
Delivery Mode : In Person
Delivery Format : Standard
Delivery Location : Kensington

General Course Information

Course Code : COMP9021
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 : In Person
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 provides students with solid conceptual knowledge and practical skills of both generic programming techniques and Python programming, to be used effectively in the many specialised courses that expect students to have acquired strong enough programming skills and well mastered the Python language. The features of the language are covered to a significant depth, and there is a strong emphasis on problem solving, from a broad base of application domains, with quite a few that involve mathematical notions. Like all foundation courses for postgraduate students, there is a lot of contents to study in limited time and the learning curve is not gentle. Still the course does not assume any prior knowledge of programming in general, or of Python programming in particular, as its content is self-contained for students with the expected mathematical background.

Course Aims

The aim of the course is to provide students with a solid foundation on fundamental programming concepts and principles, and to become proficient in the programming language Python, including object-oriented features and the tkinter library to create widgets. Students will learn to design solutions to a broad range of problems and implement those solutions in the form of small to medium programs, using appropriate programming techniques and tools.

Relationship to Other Courses

This is a core course for students enrolled in program 8545 (Master of Information Technology).

Course Learning Outcomes

Course Learning Outcomes
CLO1 : Design, implement and test programs written in a language with procedural, object-oriented, and functional constructs.
CLO2 : Apply Python language, including advanced syntax and programming techniques.
CLO3 : Analyse what happens behind the scene when operating on Python data types, with an understanding of efficiency and memory use.
CLO4 : Understand fundamental data structures and algorithms.
CLO5 : Design programs to solve small to medium scale problems.
CLO6 : Create clear, reliable, well-structured, well-tested, well-documented programs.
CLO7 : Apply appropriate tools, in particular for editing, testing and debugging.

Course Learning Outcomes	Assessment Item
CLO1 : Design, implement and test programs written in a language with procedural, object-oriented, and functional constructs.	• Assignments • Quizzes • Final Exam
CLO2 : Apply Python language, including advanced syntax and programming techniques.	• Assignments • Quizzes • Final Exam

CLO3 : Analyse what happens behind the scene when operating on Python data types, with an understanding of efficiency and memory use.	• Assignments • Quizzes • Final Exam
CLO4 : Understand fundamental data structures and algorithms.	• Assignments • Quizzes • Final Exam
CLO5 : Design programs to solve small to medium scale problems.	• Assignments • Quizzes • Final Exam
CLO6 : Create clear, reliable, well-structured, well-tested, well-documented programs.	• Assignments • Quizzes • Final Exam
CLO7 : Apply appropriate tools, in particular for editing, testing and debugging.	• Assignments • Quizzes • Final Exam

Learning and Teaching Technologies

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

Learning and Teaching in this course

The two 2-hour lectures will discuss part of the notes, some of which come with automatically generated videos. To fully benefit from the lectures, students should beforehand study the available notes and videos. The weekly lectures will also discuss quizzes and assignments as they are released. Lectures are designed to help students acquire good learning strategies, provide valuable insight, and improve problem solving skills.

Consultations/Tutorials are for individual contact, to help resolve more individual issues and get personal support for the homework, clarify concepts, get feedback, practice better.

Online discussions are for exchanges, for being part of a community, where everyone seeks support and provides support to others on any matter than is of interest to other students.

From Week 2 to Week 8 included, with Week 6 excluded, programming quizzes will be released after the second lecture and answers should be submitted by Thursday 9pm the following week (except for Quiz 4, released in Week 5 and due Thursday 9pm Week 7); details on submission will be provided during lectures. Quizzes will help students master the fundamental notions and techniques that will have been presented during lectures up to the previous week, keep up to date with the current material, and give them confidence that they are well on track.

Assignments will allow students to turn theory into practice, transform passive knowledge into active knowledge, design solutions to problems, and experience the many ways of making mistakes and correcting them when translating an algorithmic solution to an implementation. There will be two assignments, due by Monday 10am Week 7 and Week 11, respectively.

The final exam will be prac exam, taking place on campus, in CSE labs, requesting students to complete code stubs that will take advantage of the doctest module, with some tests being provided in the stubs.

Assessments

Assessment Structure

Assessment Item	Weight	Relevant Dates
Assignments Assessment FormatIndividual	26%	Start DateEach assignment is released between 3 and 4 weeks before it is due, and discussed in some of the lectures in between. Due DateThe first assignment is due by 10:00am Monday Week 7. The second assignment is due by 10:00am Monday Week 11.
Quizzes Assessment FormatIndividual	24%	Start DateEach quiz is released on Wednesday of the previous week, after the lecture has taken place and introduced the quiz. Due DateThe 6 quizzes are due by 9:00pm Thursday Weeks 3, 4, 5, 7, 8, and 9.
Final Exam Assessment FormatIndividual	50%	Start DateTBA during Exam Week Due DateTBA during Exam Week

Assessment Details

Assignments

Assessment Overview

The two assignments are programming assignments, each being worth 13%.

Sample tests are provided together with assignment specifications.

Submissions are automatically assessed against a battery of tests, all different to the sample tests.

Students are encouraged to seek syntactic, stylistic and algorithmic feedback on their implementation during consultation.

Course Learning Outcomes

- CLO1 : Design, implement and test programs written in a language with procedural, object-oriented, and functional constructs.
- CLO2 : Apply Python language, including advanced syntax and programming techniques.
- CLO3 : Analyse what happens behind the scene when operating on Python data types, with an understanding of efficiency and memory use.

- CLO4 : Understand fundamental data structures and algorithms.
- CLO5 : Design programs to solve small to medium scale problems.
- CLO6 : Create clear, reliable, well-structured, well-tested, well-documented programs.
- CLO7 : Apply appropriate tools, in particular for editing, testing and debugging.

Assessment information

Quizzes

Assessment Overview

The six quizzes are small programming assignments, each being worth 4%.

Sample tests are provided together with quiz specifications.

Submissions are automatically assessed against a battery of tests, all different to the sample tests.

Students are encouraged to study the sample solutions as well as seek syntactic, stylistic and algorithmic feedback on their implementation during consultations/tutorials.

Course Learning Outcomes

- CLO1 : Design, implement and test programs written in a language with procedural, object-oriented, and functional constructs.
- CLO2 : Apply Python language, including advanced syntax and programming techniques.
- CLO3 : Analyse what happens behind the scene when operating on Python data types, with an understanding of efficiency and memory use.
- CLO4 : Understand fundamental data structures and algorithms.
- CLO5 : Design programs to solve small to medium scale problems.
- CLO6 : Create clear, reliable, well-structured, well-tested, well-documented programs.
- CLO7 : Apply appropriate tools, in particular for editing, testing and debugging.

Final Exam

Assessment Overview

The final exam consists of independent programming questions, to be automatically assessed against a battery of tests.

Programming stubs are provided as templates that make use of the doctest module, with some tests inserted to help students develop and debug their code.

Course Learning Outcomes

- CLO1 : Design, implement and test programs written in a language with procedural, object-oriented, and functional constructs.
- CLO2 : Apply Python language, including advanced syntax and programming techniques.
- CLO3 : Analyse what happens behind the scene when operating on Python data types, with an understanding of efficiency and memory use.
- CLO4 : Understand fundamental data structures and algorithms.
- CLO5 : Design programs to solve small to medium scale problems.
- CLO6 : Create clear, reliable, well-structured, well-tested, well-documented programs.
- CLO7 : Apply appropriate tools, in particular for editing, testing and debugging.

General Assessment Information

Grading Basis

Standard

Requirements to pass course

A total mark of at least 50.

Course Schedule

Attendance Requirements

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

General Schedule Information

The following outlines a **provisional** schedule for this course. The contents of the lectures are described **roughly** and are subject to **adjustments**:

Week 1: Introduction to operators, strings, lists, tuples, dictionaries, control structures, reading from files, printing, functions.

Week 2: Functions from the random module. Exceptions. Base systems, modulo operations. Unicode character set. Sorting, lambda expressions.

Week 3: Approximation in computations. String formatting. Lists and sets, with a view on time complexity, plotting, timing. Slices, lists with a view on space complexity.

Week 4: Operations on files and directories, system operations. Default arguments. Bitwise operations. The collections and matplotlib modules.

Week 5: Special modules. Generator functions. 2-dimensional lists, numpy arrays and operations. Regular expressions.

Week 7: More special modules. Recursion. Memoisation. From recursive implementations to iterative implementations

Week 8: Classes, objects. Object-oriented programming. Special methods.

Week 9: Dynamic programming. Inheritance. Decorators.

Week 10: Searching. Sorting.

Course Resources

Prescribed Resources

There is no required textbook, and the provided material is self-contained.

Jupyter notebook sheets, together with static html files produced from those, will be provided as notes. Some of the notes are complemented with automatically produced videos. Jupyter notebook sheets offer many advantages over the more traditional lecture notes: the cells that make up a Jupyter notebook sheet can be edited, cells can be added or deleted, cells that contain code can be excuted, allowing students to guess what the output will be and check that the guess is correct, letting students play a more active role when they learn from existing code. These Jupyter notebook sheets have been very carefully designed to cover an extensive part of the Python language and include, besides all the basics, advanced syntax and programming techniques, more than found in most textbooks, all presented in the context of interesting problems, most of which should be relevant to the practical problems that will be encountered in the workplace or in other courses.

Course Evaluation and Development

This course is evaluated each term using the myExperience survey system at the end of the term.

Your feedback is important and will be considered to improve future offerings of this course. Students are also encouraged to provide informal feedback during the term, and let the lecturer and tutors know of any problems as soon as they arise. Suggestions will be listened to very openly, positively, constructively and thankfully, and every reasonable effort will be made to address them as soon as possible.

Staff Details

Position	Name	Email	Location	Phone	Availability	Equitable Learning Services Contact	Primary Contact
Convenor	Rachid Hamadi	r.hamadi@unsw.edu.au				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)