Course Outline

Contents

- Course Details
- Course Summary
- Assumed Knowledge
- Student Learning Outcomes
- Teaching Strategies
- Teaching Rationale
- Student Conduct
- Assessment
- Course Schedule
- Resources for Students
- Course Evaluation and Development
- Covid-19 Notices

Course Details

Course Code	COMP9021
Course Title	Principles of Programming
Convenor	Dr Rachid Hamadi (/users/z2286838)
Admin	Dr Rachid Hamadi (/users/z2286838)
Tutors	Matthew Perry (https://webcms3.cse.unsw.edu.au/users/z5075269) Sahil Punchhi (https://webcms3.cse.unsw.edu.au/users/z5204256) Rahma Mukta (https://webcms3.cse.unsw.edu.au/users/z5148316)
Classes	**Lectures** : Wednesday 18:00 - 20:00 Online Thursday 18:00 - 20:00 Online Timetable for all classes (/COMP9021/20T3/timetable)
Consultations	See Timetable for all classes (https://webcms3.cse.unsw.edu.au/COMP9021/20T3/timetable)
Units of Credit	6
Course Website	http://cse.unsw.edu.au/~cs9021 (https://webcms3.cse.unsw.edu.au/COMP9021/20T3/)
Handbook Entry	http://www.handbook.unsw.edu.au/postgraduate/courses/current/COMP9021.html (http://www.handbook.unsw.edu.au/postgraduate/courses/current/COMP9021.html)

Course Summary

This is a Level 0 course. It has no prerequisite. Like most Level 0 courses, it consists of bridging material in computing taught at an accelerated pace. It is a prerequisite to a number of courses that can be taken as part of Program 7543 (https://www.handbook.unsw.edu.au/postgraduate/programs/2020/7543) (Graduate Certificate in Computing), Program 5543 (https://www.handbook.unsw.edu.au/postgraduate/programs/2020/5543) (Graduate Diploma in Information Technology) and Program 8543 (https://www.handbook.unsw.edu.au/postgraduate/programs/2020/8543) (Master of Information Technology). It is a core course in the last two programs. Students from other Faculties and Schools, enrolled in either coursework or research degrees and who have an interest in, or a need for, programming skills, especially in Python, can also enrol in the course.

The aim of the course is to provide students with a solid foundation on fundamental programming concepts and principles, develop problem solving skills, and master the programming language Python.

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.

Assumed Knowledge

Though there is no assumed knowledge, **mathematical maturity** and **familiarity with some form of programming** might reduce the amount of time and efforts that will have to be dedicated to the course, as the **learning curve is steep** and **programming can be very time consuming** , especially for **beginners** .

Student Learning Outcomes

After successfully completing this course, students will be able to:

1. Know how to design, implement and test programs written in a language with procedural, object-oriented, and functional constructs.
2. Be proficient in the Python language, including advanced syntax and programming techniques.
3. Gain insights on what happens behind the scene when operating on Python data types, with an understanding of efficiency and memory use.
4. Have a first acquaintance with fundamental data structures and algorithms.
5. Know how to design programs to solve small to medium scale problems.
6. Be able to write clear, reliable, well-structured, well-tested, well-documented programs.
7. Be proficient in the use of appropriate tools, in particular for editing, testing and debugging.
8. Know how to plot data in various ways, record animation movies.
9. Be exposed to a variety of problems related to more specialised fields and taught in other courses (Turing machines, k-clustering, Prolog, Nash equilibrium, cryptography, fractals, ...).

This course contributes to the development of the following graduate capabilities:

Graduate Capability	Acquired in
Scholars capable of independent and collaborative enquiry, rigorous in their analysis, critique and reflection, and able to innovate by applying their knowledge and skills to the solution of novel as well as routine problems	Practice exercises, Quizzes, Assignments
Entrepreneurial leaders capable of initiating and embracing innovation and change, as well as engaging and enabling others to contribute to change	Practice exercises, Quizzes, Assignments

Professionals capable of ethical, self- directed practice and independent lifelong learning	Practice exercises, Quizzes, Assignments
Global citizens who are culturally adept and capable of respecting diversity and acting in a socially just and responsible way	Practice exercises, Quizzes, Assignments

Teaching Strategies

- **Lectures** : Use problem solving to present the material; these are designed to help acquire good learning strategies and provide valuable insight. Lectures will be online, live and recorded with Blackboard Collaborate Ultra through **Moodle** on the days and times lectures have been timetabled. For instructions on how to see the lectures live and access the recordings later, see Blackboard Collaborate Ultra (https://student.unsw.edu.au/blackboard-collaborate-ultra) . You will also be able to access (from COMP9021 Moodle shell) and watch the recorded lectures any time after they become available, but attending the lecture live gives you the option to interact with the lecturer and most importantly, ask questions.
- **Consultations** : Help resolve more individual issues and get personal support for the homework, clarify concepts, get feedback, practice better. There will be five 2-hour consultations (see Timetable for all classes (https://webcms3.cse.unsw.edu.au/COMP9021/20T3/timetable)). They will be run by enthusiastic, very helpful, and very competent tutors using Blackboard Collaborate Ultra through **Moodle** .
- **Online forums discussions** : For exchanges, being part of a community, where everyone seeks support and provides support to others on any matter that is of interest to other students. We will be using **Ed** platform for forums discussions.
- **Coding quizzes** : Let you to master the fundamental concepts and techniques that will have been presented during lectures up to the previous week, keep up to date with the current material, and give you confidence that you are well on track. You will be submitting your coding quizzes in **Ed** platform.
- **Assignments** : Let you 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. You will also be submitting your assignments in **Ed** platform.

Teaching Rationale

You know that, at university, the focus is on your self-directed search for knowledge. Lectures, consultations, online discussions, textbook and recommended readings, coding quizzes, practice exercises, assignments, and final exam are all provided as a service to assist you in this endeavour. It is your choice as to how much work you do in this course, whether it is preparation for classes, completion of coding quizzes and assignments, study for final exam, seeking assistance, or extra work to extend and clarify your understanding. You must choose the approach that best suits your learning style and goals in this course. Please note that the University expects you to spend about 150 hours work for this course (including attending lectures as well as time spent on self-study and doing coding quizzes and assignments). Of course this may vary according to your aims.

The course is designed in such a way that passing the course will only require a sufficient understanding of the fundamental programming concepts as well as decent practical skills, thanks to consistent and regular work. If your aim is to obtain High Distinction then you will need to invest more time in this course.

Student Conduct

The **Student Code of Conduct** (Information (https://student.unsw.edu.au/conduct) , Policy (https://www.gs.unsw.edu.au/policy/documents/studentcodepolicy.pdf)) sets out what the University expects from students as members of the UNSW community. As well as the learning, teaching and research environment, the University aims to provide an environment that enables students to achieve their full potential and to provide an experience consistent with the University's values and guiding principles. A condition of enrolment is that students *inform themselves* of the University's rules and policies affecting them, and conduct themselves accordingly.

In particular, students have the responsibility to observe standards of equity and respect in dealing with every member of the University community. This applies to all activities on UNSW premises and all external activities related to study and research. This includes behaviour in person as well as behaviour on social media, for example Facebook groups set up for the purpose of discussing UNSW courses or course work. Behaviour that is considered in breach of the Student Code Policy as discriminatory, sexually inappropriate, bullying, harassing, invading another's privacy or causing any person to fear for their personal safety is serious misconduct and can lead to severe penalties, including suspension or exclusion from UNSW.

If you have any concerns, you may raise them with your lecturer, or approach the School Ethics Officer (mailto:ethics-officer@cse.unsw.edu.au) , Grievance Officer (mailto:grievance-officer@cse.unsw.edu.au) , or one of the student representatives.

Plagiarism is defined as (https://student.unsw.edu.au/plagiarism) using the words or ideas of others and presenting them as your own. UNSW and CSE treat plagiarism as academic misconduct, which means that it carries penalties as severe as being excluded from further study at UNSW. There are several on-line sources to help you understand what plagiarism is and how it is dealt with at UNSW:

- Plagiarism and Academic Integrity (https://student.unsw.edu.au/plagiarism)
- UNSW Plagiarism Procedure (https://www.gs.unsw.edu.au/policy/documents/plagiarismprocedure.pdf)

Make sure that you read and understand these. Ignorance is not accepted as an excuse for plagiarism. In particular, you are also responsible that your assignment files are not accessible by anyone but you by setting the correct permissions in your CSE directory and code repository, if using. Note also that plagiarism includes paying or asking another person to do a piece of work for you and then submitting it as your own work.

UNSW has an ongoing commitment to fostering a culture of learning informed by academic integrity. All UNSW staff and 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.

If you haven't done so yet, please take the time to read the full text of

- UNSW's policy regarding academic honesty and plagiarism (https://student.unsw.edu.au/plagiarism)

The pages below describe the policies and procedures in more detail:

- Student Code Policy (https://www.gs.unsw.edu.au/policy/documents/studentcodepolicy.pdf)
- Student Misconduct Procedure (https://www.gs.unsw.edu.au/policy/documents/studentmisconductprocedures.pdf)
- Plagiarism Policy Statement (https://www.gs.unsw.edu.au/policy/documents/plagiarismpolicy.pdf)
- Plagiarism Procedure (https://www.gs.unsw.edu.au/policy/documents/plagiarismprocedure.pdf)

You should also read the following page which describes your rights and responsibilities in the CSE context:

- Essential Advice for CSE Students (https://www.engineering.unsw.edu.au/computer-science-engineering/about-us/organisational-structure/student-services/policies/essential-advice-for-cse-students)

Assessment

Item	Topics	Due	Marks	Contributes to
Coding Quizzes	Topic of the week	Friday Weeks 3, 4, 5, 7, 8, and 9 @ 10pm	20%	CLOs 1-7
Assignment 1	Procedural programming	Monday Week 7 @ 10pm	15%	CLOs 1-7
Assignment 2	Object oriented programming	Sunday Week 10 @ 10pm	15%	CLOs 1-7
Final exam	Everything covered in the course	Exam period	50%	CLOs 1-7

Course Schedule

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

Week	Lectures	Assignments	Quizzes
1	Introduction to operators, lists, tuples, dictionaries, control structures, reading from files, printing, functions. Functions from the random module, exceptions.		
2	Base systems, modulo operations. Unicode character set. Sorting, lambda expressions.		
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.		Quiz 1 due Friday 10:00pm
4	Operations on files and directories, system operations. Default arguments. Bitwise operations. The collections and matplotlib modules.		Quiz 2 due Friday 10:00pm
5	Special modules. Generator functions. 2-dimensional lists, numpy arrays and operations. Regular expressions.		Quiz 3 due Friday 10:00pm
6			
7	More special modules. Recursion, memoisation. From recursive implementations to iterative implementations	Assignment 1 due Monday 10:00pm	Quiz 4 due Friday 10:00pm

8	Classes, objects. Object-oriented programming. Special methods.	Quiz 5 due Friday 10:00pm
9	Dynamic programming. Inheritance. Decorators.	Quiz 6 due Friday 10:00pm
10	Searching. Sorting.	Assignment 2 due Sunday 10:00pm

Resources for Students

For easy introductions to Python, the following texts are recommended:

- John Zelle: Python Programming: An Introduction to Computer Science (https://fbeedle.com/our-books/23-python-programming-an-introduction-to-computer-science-3rd-ed-9781590282755.html)
- Brad Miller and David Ranum: Problem Solving with Algorithms and Data Structures Using Python (https://runestone.academy/runestone/static/pythonds/index.html)
- Allen B. Downey: How to think like a computer scientist: Learning with Python (https://runestone.academy/runestone/static/thinkcspy/index.html)

For students with a good knowledge of Python already, the following texts are recommended:

- Luciano Ramalho: Fluent Python (http://shop.oreilly.com/product/0636920032519.do)
- David Beazley and Brian K. Jones: Python Cookbook (http://shop.oreilly.com/product/0636920027072.do)

Official references are richer and often invaluable:

- The Python Tutorial (https://docs.python.org/3.8/tutorial/)

They also offer the most complete coverage of the language:

- The Python Standard Library (https://docs.python.org/3.8/library/index.html)

Every week, there will be a widget, but to understand all aspects of their code, some resources are necessary. The following official reference does the job perfectly:

- Graphical User Interfaces with Tk (https://docs.python.org/3/library/tk.html)

Course Evaluation and Development

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

In the previous offerings of this course, feedback was very good. Some students suggested to give more weight to assignments. This advice will be followed. As a consequence, there will be a smaller weight on the final exam.

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 in charge 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.

Covid-19 Notices

Safe Return to Campus FAQs (https://www.covid-19.unsw.edu.au/safe-return-campus-faqs)

UNSW Engineering Covid-19 Communication (https://edtech.eng.unsw.edu.au/c19mess/comms.html)

Resource created 10 months ago (Saturday 05 September 2020, 11:23:52 AM), last modified 5 minutes ago (Friday 25 June 2021, 11:36:28 AM).

Comments

(/COMP9021/20T3/forums/search?forum_choice=resource/52214)

(/COMP9021/20T3/forums/resource/52214)

There are no comments yet.