# 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


## Course Details

| | |
|---|---|
| **Course Code** | COMP1010 |
| **Course Title** | The Art of Computing |
| **Convenor** | Robert Clifton-Everest (/users/z3254687) |
| **Admin** | Robert Clifton-Everest (/users/z3254687) |
| **Classes** | **Lectures** : Tuesday 1PM-3PM, Thursday 4-6PM<br>Timetable for all classes (/COMP1010/20T3/timetable) |
| **Consultations** | TBD |
| **Units of Credit** | 6 |
| **Course Website** | http://cse.unsw.edu.au/~cs1010/21T1/ (http://cse.unsw.edu.au/~cs1010/20T3/) |
| **Handbook Entry** | http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP1010.html (http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP1010.html) |

## Course Summary

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.

# Assumed Knowledge

Before commencing this course, students should:

- have basic computer skills (using a mouse and keyboard, browsing the web, etc)

# Student Learning Outcomes

After completing this course, students will:

1. Be able to use a spreadsheet for simple data management tasks.
2. Be able to write Python programs to solve simple computational problems.
3. Be able to solve problems via computer systems.
4. Be able to build simple human-centered interfaces to computers.

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 | Labs & Project |
| **Entrepreneurial leaders** capable of initiating and embracing innovation and change, as well as engaging and enabling others to contribute to change | Project |
| **Professionals** capable of ethical, self- directed practice and independent lifelong learning | Labs & Project |
| **Global citizens** who are culturally adept and capable of respecting diversity and acting in a socially just and responsible way | All work |

# Teaching Strategies

- Lectures ... introduce concepts, show examples
- Tutorials ... reinforce concepts and provide additional examples
- Lab Work ... provide essential practice in programming
- Project .. allows students to solve a significant problem that matters to them

# Teaching Rationale

As this course was first taught during an uncertain period in human history (global pandemic), it designed to be delivered in a flexible online-first manner. Rather than trying to simulate face-to-face teaching in an online environment, the course tries to take advantage of the benefits the new medium offers. Lectures use minimal slides and mostly focus on working through examples, with lots of opportunities to ask questions and engage with the lecturer and other students. Tutorials are similar, but in smaller groups. Lab sessions give you the opportunity to receive individual help.

# 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 |
|---|---|---|---|---|
| Project | HCI, Python programming | Weeks 7 and 10 | 30% | 2,3,4 |

| | | | | |
|---|---|---|---|---|
| Labs | All topics | Weeks 1-5, 8, and 9 | 20% | 1,2,3,4 |
| Final Exam | All topics | Exam period | 50% | 1,2,3,4 |

### Lab Exercises

In weeks 1-5, 8 and 9 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.

### Project

In the project, you will implement a web application 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 and does not necessarily have to conform to a standard web app. 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.

### Exam

The final exam will be open book and online. You will have either 12 or 24 hours to complete it, depending on timetabling during the exam period. It will involve programming exercises, but may also include multiple choice and short answer. You will **NOT** be remotely monitored while taking the exam.

The exam mark is subject to scaling and has a hurdle of 25/50 based on the scaled mark. In the event your final mark is greater than 50, but your exam mark is under 25, your final grade will be UF.

## Course Schedule

This schedule is HIGHLY subject to change based on student feedback throughout the term.

| Week | Lectures | Tutes | Labs | Project | Notes |
|---|---|---|---|---|---|
| 1 | Course intro, thinking computationally, spreadsheets, intro to python | - | - | - | - |
| 2 | Control-flow, Data, Lists | - | - | - | - |
| 3 | Data structures, Program composition | - | - | - | - |
| 4 | Algorithms | - | - | - | - |
| 5 | Networks and the web (HTML, CSS, etc.) | - | - | - | - |
| 6 | Revision & Practice | - | - | - | Flexibility week |
| 7 | Human-Computer Interaction, User Interfaces | - | - | - | - |
| 8 | Testing | - | - | - | - |
| 9 | Advanced topics | - | - | - | - |

| | | | | | |
|---|---|---|---|---|---|
| 10 | Revision and review | - | - | - | - |

# Resources for Students

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/ (https://greenteapress.com/wp/think-python-2e/)

# Course Evaluation and Development

Students last term indicated they would have liked more challenge exercises, so additional challenge exercises will be added to the labs each week.

As the course is still fairly young, it is important that the teaching staff learn about the students in the course. As such, there will be a survey at the start of the course to establish what students want from the course, as well as their background and experience. A similar survey will be run halfway through the course.

Resource created 5 months ago (Saturday 30 January 2021, 04:15:57 PM), last modified 5 months ago (Saturday 30 January 2021, 04:44:55 PM).

Comments

(/COMP1010/21T1/forums/search?forum_choice=resource/56897)

(/COMP1010/21T1/forums/resource/56897)

There are no comments yet.