UNSW Course Outline

CODE1161 Design Computing - 2024

Published on the 21 May 2024

General Course Information

Course Code : CODE1161
Year : 2024
Term : Term 2
Teaching Period : T2
Is a multi-term course? : No
Faculty : Faculty of Arts, Design and Architecture
Academic Unit : School of Built Environment
Delivery Mode : In Person
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

Design Computing introduces you to the exploration and visualisation of complex data. Using a synthesis of technical and theoretical knowledge of design computing, you will apply skills in a text-based programming language to inform your own design projects and professional work.

You will participate in weekly activities to experiment with and develop your technical knowledge. You will also develop your verbal and digital communication skills to present explorations and results.

Course Learning Outcomes

Course Learning Outcomes
CLO1 : Synthesise computing theory and technical knowledge for design.
CLO2 : Apply design computing in your own design projects and professional work.
CLO3 : Apply programming skills in a text-based language.
CLO4 : Communicate complex design computing data through scripts and digital images.

Course Learning Outcomes	Assessment Item
CLO1 : Synthesise computing theory and technical knowledge for design.	• Assignment • Assignment
CLO2 : Apply design computing in your own design projects and professional work.	• Assignment • Assignment
CLO3 : Apply programming skills in a text-based language.	• Assignment • Assignment
CLO4 : Communicate complex design computing data through scripts and digital images.	• Assignment

Learning and Teaching Technologies

Moodle - Learning Management System | Microsoft Teams | Echo 360

Assessments

Assessment Structure

Assessment Item	Weight	Relevant Dates
Assignment Assessment Format: Individual Short Extension: Yes (7 days)	60%	Start Date: Not Applicable Due Date: Week 7: 08 July - 14 July
Assignment Assessment Format: Individual Short Extension: Yes (4 days)	40%	Start Date: Not Applicable Due Date: Week 12: 12 August - 18 August

Assessment Details

Assignment

Assessment Overview

You will select and explore an open dataset and design a way to visually represent and effectively communicate your insights. You will create an open data visualisation using text-based programming. Grading will be done against assessment criteria accompanied by written feedback. Verbal feedback will also be given in tutorials.

Course Learning Outcomes

- CLO1 : Synthesise computing theory and technical knowledge for design.
- CLO2 : Apply design computing in your own design projects and professional work.
- CLO3 : Apply programming skills in a text-based language.
- CLO4 : Communicate complex design computing data through scripts and digital images.

Detailed Assessment Description

1. **Weekly programming exercises (weeks 1-5)**

The weekly programming exercises are tasks that explore and reinforce the concepts covered in the lectures and labs.
Each week until week 5 there will be tasks given to students during the lecture and tutorial face-to-face sessions. These exercises are to be uploaded to GitHub and will be marked automatically, but the code used to mark them is available to you so you can test if your work will pass or not.

Criteria / Deliverables:

Each week at the cut-off time (7pm the following Tuesday) each student's GitHub repository will be read by a script and the code present will be marked. It will either pass or fail the tests.

Feedback will also be available through the discussion forum and with a tutor during the tutorial.

1. **Quiz (week 7 in-class)**

The quiz will be conducted through GitHub and will consist of a series of small, simple programming problems. The subject of the questions will be given ahead of time, but the specifics will be revealed on the day of the quiz.

Criteria / Deliverables:

This quiz will test a student's ability to apply basic programming principles to set problems.

Assignment submission Turnitin type

Not Applicable

Assignment

Assessment Overview

You will apply skills in a text-based programming language to inform and develop your own design project. Grading will be done against assessment criteria accompanied by written feedback. Verbal feedback will also be given in tutorials.

Course Learning Outcomes

- CLO1 : Synthesise computing theory and technical knowledge for design.
- CLO2 : Apply design computing in your own design projects and professional work.
- CLO3 : Apply programming skills in a text-based language.

Detailed Assessment Description

1. **Open data project (week 11)**

This assignment explores how to make data accessible to everyone. This is the capstone project for this course. You will take an open data set, from a collection such as the NSW government and build a way to represent it to others. You may use any available libraries or frameworks to do so, although matplotlib is recommended.

Criteria: Refer to rubric on Moodle
Deliverables: A presentation and demonstration of your data exploration tool in week 11's class.

1. **Submission (week 11)**
   Your GitHub Repository will be marked at its state at 7pm Tuesday of week 12.
2. **Git collaboration (week 11)**

Working together to achieve greatness, through GitHub collaboration. One of the greatest things about being a student is having a cohort to go through the struggle of your courses with. One of the greatest things about open-source culture is that there is a way for you to share the burden by helping each other out. This is a mark that is available to anyone who makes a Pull Request to

anyone else’s repository during the Open Data Project and has it accepted.

Deliverables: Proof of an accepted pull request, in both directions, described in your lab book.

Assignment submission Turnitin type

Not Applicable

General Assessment Information

Grading Basis

Standard

Course Schedule

Teaching Week/Module	Activity Type	Content
Week 0 : 20 May - 26 May	Activity	No class this week, but you can use this time to make sure that your computer is up to date, and you know how to use it. It might be helpful for you to look at this course: https://swcarpentry.github.io/shell-novice/
Week 1 : 27 May - 2 June	Lecture	Programming, an introduction An introduction to programming concepts, its impact on the world and how you can be a part of extending that impact. week will contain a lot of information about how the course will function, how to stay healthy as a programmer, and the basics of Python syntax. Optional Reading: • Graham, P. (2009). Maker's Schedule, Manager's Schedule. • Case, N. (2016). Simulating The World (In Emoji ????). • Davis, D. (2015). Why Architects Can't Be Automated. • Doherty, B. (2015). Architects getting automated?
	Tutorial	Getting your environment set up
Week 2 : 3 June - 9 June	Lecture	Programming on your own terms This week you'll be working in your own environment. This is the beginning of a process of refinement that will last for your whole career. We'll talk about abstraction and problem decomposition. How to break up problems into small enough chunks to solve, and how to make those solutions as general as possible. You'll use tests to make sure that your code what you think it does. Optional Reading: • Victor, B. (2011). Up and Down the Ladder of Abstraction. • Doherty, B. (2016). A thinking trick for beginner programmers - Ghetto TDD • Noll, A. M. (1967). The digital computer as a creative medium. IEEE Spectrum, 4(10), 89–95.
	Tutorial	
Week 3 : 10 June - 16 June	Lecture	Introduction to algorithms Now that you can write simple code we are going to start putting it to work. We'll learn to implement simple algorithms, and how to compare different ways of doing the same thing using time complexity. Required Reading: Cormen, T. & Balkcom, D., Algorithms. Go through the Intro to algorithms, Binary search and Asymptotic notation sections
	Tutorial	
Week 4 : 17 June - 23 June	Lecture	Python data-science tools Python is one of the most commonly used scientific programming languages. It has comprehensive libraries for almost anything you can imagine needing to do to data. We'll go over Matplotlib and Pandas. We'll also cover data cleaning in a repeatable way. Visualising data is very fashionable, there are lots of whizzy ways to do this, but we'll go over the absolute basics of data-vis. Required Reading: • Polich, K. (2016). Let's Kill the Word Cloud. Optional Reading: • Tufte, E. (2001) The Visual Display of Quantitative Information Read chapter 4: Data-Ink and Graphical Redesign. If you feel so inclined, read the whole book! Wong, D. (2010) • The Wall Street Journal Guide to Information Graphics: The Dos and Don'ts of Presenting Data, Facts, and Figures. Refactoring Sooner or later you'll want to work with other people on a project. If you have a monolithic block of code this will be very difficult. Well refactored code is easy to maintain and easier to share with others. Optional Reading: • The Catalog of Refactoring. refactoring.guru. Have a look around this site for inspiration.
	Tutorial	
Week 5 : 24 June - 30 June	Lecture	Input/Output (IO) Programs that are completely self-contained are never going to be all that useful. This week we'll learn how to read and write files and get information from the internet. We'll also go over the history of computing in the world, and in architecture. Optional Reading: • Victor, B. (2013). The Future of Programming. • Polich, K. (2016). Potholes. • Polich, K. (2015). NYC Speed Camera Analysis.
Week 6 : 1 July - 7 July	Activity	No scheduled class --flexibility week

Week 7 : 8 July - 14 July	Lecture	Python in other places, QUIZ Python is often used as a scripting language. It can be used to control Grasshopper and Dynamo. We'll go over how to do this, and how to find out what is available in the environment that you find yourself in. There will also be a programming exam to test recall on basic principles. The subject of the questions will be given ahead of time, but the specifics will only be revealed on the day. Optional Reading: • Victor, B. Inventing On Principle. worrydream.com/#!/InventingOnPrinciple • Samuel, A. (2015). How to Give a Data-Heavy Presentation.
	Tutorial	Quiz This will be like a regular week's exercises. The only difference is that it's timed and we'll do it in class.
Week 8 : 15 July - 21 July	Lecture	Tricks & patterns This lecture will cover some useful python ideas that don't fit in anywhere else. Regex, list comprehensions, slicing. We'll also cover touch on the basics of data ethics. Required Reading: • Angwin, J., Larson, J., Mattu, S. and Kirchner , L., ProPublica (2016) Machine Bias Optional Reading: • Floridi, L., & Taddeo, M. (2016). What is data ethics? Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 374(2083). • Urban, T. (2015). The Artificial Intelligence Revolution: Part 1 - Wait But Why. • Urban, T. (2015). The Artificial Intelligence Revolution: Part 2 - Wait But Why. • M. Sanderson, A. Sandberg (2016). Battle Cry - Anders Sandberg on ethical AI
	Tutorial	
Week 9 : 22 July - 28 July	Lecture	Flexible subject week We'll choose a subject for this week depending on the interests of the class.
Week 10 : 29 July - 4 August	Lecture	Fitting it all together At this stage you will know a lot of slightly disjointed things about python programming. This lecture will tie up the loose ends and help you transition into using your skills in 'real' contexts. MyExperience
	Tutorial	In class tutorials covering Assignment 3
Week 11 : 5 August - 11 August	Activity	No class this week, time for independent study
Week 12 : 12 August - 18 August	Presentation	Assignment three presentations

Attendance Requirements

You are expected to be regular and punctual in attendance at all classes for the School of Built Environment courses in which you are enrolled. If and where individual courses have specific attendance requirements, these will be stated in the course outline.

If you do not attend, engage, or participate in scheduled class activities, including lectures, tutorials, studios, labs, etc, you run the risk of failing a course.

If illness or unexpected and beyond your control circumstances prevent you from completing a task on time, or substantially disturb your assessment performance, you should apply for Special Consideration, as soon as practicable, accompanied by appropriate documentation.

No special consideration will be provided if you miss out on essential course information and materials, or if you miss assessment tasks and deadlines due to unexplained absences or an unapproved lack of attendance.

You may be advised by the Course Convenor to withdraw from the course if significant learning activities are missed.

Course Resources

Course Evaluation and Development

Each student has the opportunity to participate in the MyExperience course survey for CODE1161 at the end of the term. This is an anonymous questionnaire in which students can register their level of satisfaction with the content, delivery, teaching and management of the course. Secondly, students are welcome to make an appointment at any point throughout the term to meet with the course convenor to discuss any issues they encounter within the course. These meetings are private and confidential.

Staff Details

Position	Name	Email	Location	Phone	Availability	Equitable Learning Services Contact	Primary Contact
Convenor	Ben Doherty				Always available via Teams, will reply between 10 and 4 on weekdays.	No	Yes

Other Useful Information

Academic Information

Due to evolving advice by NSW Health, students must check for updated information regarding online learning for all Arts, Design and Architecture courses this term (via Moodle or course information provided).

Please see: https://www.unsw.edu.au/arts-design-architecture/student-life/resources-support/protocols-guidelines for essential student information relating to:

- UNSW and Faculty policies and procedures;
- Student Support Services;
- Dean's List;
- review of results;
- credit transfer;
- cross-institutional study and exchange;
- examination information;
- enrolment information;
- Special Consideration in the event of illness or misadventure;

- student equity and disability;

And other essential academic information.

Academic Honesty and Plagarism

Plagiarism is using the words or ideas of others and presenting them as your own. It can take many forms, from deliberate cheating to accidentally copying from a source without acknowledgement.

UNSW groups plagiarism into the following categories:

- Copying: Using the same or very similar words to the original text or idea without acknowledging the source or using quotation marks. This includes copying materials, ideas or concepts from a book, article, report or other written document, presentation, composition, artwork, design, drawing, circuitry, computer program or software, website, internet, other electronic resource, or another person's assignment without appropriate acknowledgement.
- Inappropriate paraphrasing: Changing a few words and phrases while mostly retaining the original information, structure and/or progression of ideas of the original without acknowledgement. This also applies in presentations where someone paraphrases another's ideas or words without credit and to piecing together quotes and paraphrases into a new whole, without appropriate referencing.
- Collusion: Working with others but passing off the work as a person's individual work. Collusion also includes providing your work to another student for the purpose of them plagiarising, paying another person to perform an academic task, stealing or acquiring another person's academic work and copying it, offering to complete another person's work or seeking payment for completing academic work.
- Inappropriate citation: Citing sources which have not been read, without acknowledging the "secondary" source from which knowledge of them has been obtained.
- Duplication ("self-plagiarism"): Submitting your own work, in whole or in part, where it has previously been prepared or submitted for another assessment or course at UNSW or another university.

The UNSW Academic Skills support offers resources and individual consultations. Students are also reminded that careful time management is an important part of study. One of the identified causes of plagiarism is poor time management. Students should allow sufficient time for research, drafting and proper referencing of sources in preparing all assessment items. UNSW Library has the ELISE tool available to assist you with your study at UNSW. ELISE is designed to introduce new students to studying at UNSW, but it can also be a great refresher during your study.

Completing the ELISE tutorial and quiz will enable you to:

- analyse topics, plan responses and organise research for academic writing and other assessment tasks
- effectively and efficiently find appropriate information sources and evaluate relevance to your needs
- use and manage information effectively to accomplish a specific purpose
- better manage your time
- understand your rights and responsibilities as a student at UNSW
- be aware of plagiarism, copyright, UNSW Student Code of Conduct and Acceptable Use of UNSW ICT Resources Policy
- be aware of the standards of behaviour expected of everyone in the UNSW community
- locate services and information about UNSW and UNSW Library

Use of AI for assessments

As AI applications continue to develop, and technology rapidly progresses around us, we remain committed to our values around academic integrity at UNSW. Where the use of AI tools, such as ChatGPT, has been permitted by your course convener, they must be properly credited and your submissions must be substantially your own work.

In cases where the use of AI has been prohibited, please respect this and be aware that where unauthorised use is detected, penalties will apply.

Use of AI for assessments | UNSW Current Students

Submission of Assessment Tasks

Turnitin Submission

If you encounter a problem when attempting to submit your assignment through Turnitin, please telephone External Support on 9385 3331 or email them on externalteltsupport@unsw.edu.au

Support hours are 8:00am – 10:00pm on weekdays and 9:00am – 5:00pm on weekends (365 days a year). If you are unable to submit your assignment due to a fault with Turnitin, you may apply for an extension, but you must retain your ticket number from External Support (along with any other relevant documents) to include as evidence to support your extension application. If you email External Support, you will automatically receive a ticket number, but if you telephone, you will need to specifically ask for one. Turnitin also provides updates on their system status on Twitter.

Generally, assessment tasks must be submitted electronically via either Turnitin or a Moodle

assignment. In instances where this is not possible, alternative submission details will be stated on your course's Moodle site. For information on how to submit assignments online via Moodle: https://student.unsw.edu.au/how-submit-assignment-moodle

Late Submission Penalty

UNSW has a standard late submission penalty of:

- 5% per calendar day,
- for all assessments where a penalty applies,
- capped at five calendar days (120 hours) from the assessment deadline, after which a student cannot submit an assessment, and
- no permitted variation.

Students are expected to manage their time to meet deadlines and to request Special Consideration as early as possible before the deadline. Support with Time Management is available here.

School Contact Information

beadmin@unsw.edu.au