Built Environment

CODE1161
Design Computing

Ben Doherty
Disclaimer
This abbreviated course outline is indicative of the outcomes, delivery and assessment. While Course Learning Outcomes will remain constant, other details may be subject to change. The full and most accurate course outline will be available in Moodle.

1. COURSE STAFF

<table>
<thead>
<tr>
<th>Course Convenor</th>
<th>Ben Doherty</th>
</tr>
</thead>
<tbody>
<tr>
<td>Email</td>
<td><a href="mailto:b.doherty@unsw.edu.au">b.doherty@unsw.edu.au</a></td>
</tr>
</tbody>
</table>

2. COURSE DETAILS

Credit Points: 6 UoC

<table>
<thead>
<tr>
<th>Learning Activity</th>
<th>Hours per week</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td>1</td>
</tr>
<tr>
<td>Studio</td>
<td>4</td>
</tr>
<tr>
<td>Online learning activity</td>
<td>1</td>
</tr>
</tbody>
</table>

Description

The communication and visualisation course introduces students to the simulation, manipulation and visualisation of complex data information. Using a synthesis of technical and theoretical knowledge of design computing, students will apply skills in a text based programming language to inform their own design projects and professional work. Students will participate to weekly laboratory-based activities to experiment with and develop further their technical knowledge. The students performance will be evaluated with respect to their progressive work developed in preparation for or during the studio activities. Students will employ a range of verbal and multimedia communication skills to demonstrate their explorations and results.

Program Learning Outcomes (PLOs)

The Program Learning Outcomes from Architectural Studies addressed in this course are:

1. Synthesise interdisciplinary knowledge of cultural, natural, and technological systems in local and global contexts.
2. Apply interdisciplinary knowledge using computational design thinking and methods to built environment challenges.
3. Critically analyse complex environmental conditions through digital technologies and computational methods.
4. Apply computational design knowledge and skills for professional work and, or further learning.
5. Practice the ethical application of digital and computational technologies in and for the design of the built environment.

Course Learning Outcomes (CLOs) with Alignment to PLOs and Assessment

<table>
<thead>
<tr>
<th>CLO #</th>
<th>CLO Statement</th>
<th>PLO #</th>
<th>Related Assessment</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLO 1</td>
<td>Demonstrate a synthesis of technical and theoretical knowledge of design computing</td>
<td>1</td>
<td>Assessment 1 / 2 / 3</td>
</tr>
<tr>
<td>CLO 2</td>
<td>Apply design computing in their own design projects and professional work.</td>
<td>2, 4</td>
<td>Assessment 1 / 2 / 3</td>
</tr>
<tr>
<td>CLO 3</td>
<td>Apply skills in a text based programming language</td>
<td>4</td>
<td>Assessment 1 / 2 / 3</td>
</tr>
<tr>
<td>CLO 4</td>
<td>Demonstrate relevant verbal and multimedia communication skills.</td>
<td>4</td>
<td>Assessment 1 / 2 / 3</td>
</tr>
</tbody>
</table>
3. ASSESSMENT

<table>
<thead>
<tr>
<th>Assessment Task</th>
<th>Weight</th>
<th>Course Learning Outcomes Assessed</th>
<th>Due Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Assignment (Lab Book, Weekly Programming Exercise, Exam)</td>
<td>50%</td>
<td>CLO 1 / CLO 2 / CLO 3 / CLO 4</td>
<td>Week 1-10 / 3, 6, 10</td>
</tr>
<tr>
<td>2. Assignment (Holy Wars)</td>
<td>20%</td>
<td>CLO 1 / CLO 4</td>
<td>Week 6</td>
</tr>
<tr>
<td>3. Assignment (Open Data Project / GIT collaboration)</td>
<td>30%</td>
<td>CLO 1 / CLO 2 / CLO 3 / CLO 4</td>
<td>Week 10</td>
</tr>
</tbody>
</table>

4. WEEKLY COURSE SCHEDULE

<table>
<thead>
<tr>
<th>Week</th>
<th>Topic</th>
<th>Activity</th>
<th>Related CLO</th>
</tr>
</thead>
</table>
| Week 1 | Programming, an introduction       | • The first lecture is an introduction to programming concepts, its impact on the world and how you can be a part of extending that impact. This 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.  
  • In class tutorials covering Assignment 1.                                                                                     | CLO 1  
  CLO 2  
  CLO 3  |
| Week 2 | Programming on your own terms      | • This week Students will be working in their own environment. This is the beginning of a process of refinement that will last for your whole career. The lecture will 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. Students will use tests to make sure that your code what you think it does.  
  • In class tutorials covering Assignment 1.                                                                                     | CLO 1  
  CLO 2  
  CLO 3  |
| Week 3 | Introduction to algorithms        | • Now that you can write simple code we are going to start putting it to work. Students will learn to implement simple algorithms, and how to compare different ways of doing the same thing using time complexity.  
  • Presentation Assignment 1.                                                                                                      | CLO 1  
  CLO 4  |
| Week 4 | IO                                 | • Programs that are completely self-contained are never going to be all that useful. In this week lecture students will 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 in particular.  
  • In class tutorials covering Assignment 2.                                                                                     | CLO 1  
  CLO 4  |
| Week 5   | Refactoring and Introduction to AI | • 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. The second part will introduce AI (Artificial intelligence) a topic which is a huge, but it’s often poorly defined. This lecture will be a very broad introduction to some definitions of what it means.  
• In class tutorials covering Assignment 2. | CLO 1  
CLO 4 |
| Week 6   | Working with data | • 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. In this lecture we will introduce 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.  
• Presentation Assignment 2. | CLO 1  
CLO 4 |
| Week 7   | Tricks, patterns and ethics | • This lecture will cover some useful python ideas that don't fit in anywhere else such as regex, list comprehensions, slicing. The second part of the lecture also cover touch on the basics of data ethics.  
• In class tutorials covering Assignment 3. | CLO 1  
CLO 2  
CLO 3 |
| Week 8   | Python in x | • 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.  
• In Class exam.  
• In class tutorials covering Assignment 3. | CLO 1  
CLO 2  
CLO 3 |
| Week 9   | 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.  
• In class tutorials covering Assignment 3. | CLO 1  
CLO 2  
CLO 3 |
| Week 10  | Final Presentation | • Presentation Assignment 3. | CLO 1  
CLO 4 |