ENGG1811 Computing for Engineers

Course Introduction
- Staff
- Course Objectives
- Ways of Learning
- Lecture Schedule
- Assessment
- Avoiding Plagiarism
- Software

Course Website:
https://webcms3.cse.unsw.edu.au/ENGG1811/19T2/
# People and Website

<table>
<thead>
<tr>
<th>Staff Name</th>
<th>Role</th>
<th>Email (Best method)</th>
<th>Phone (ext)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ashesh Mahidadia</td>
<td>Convenor and Lecturer</td>
<td><a href="mailto:ashesh@cse.unsw.edu.au">ashesh@cse.unsw.edu.au</a></td>
<td></td>
</tr>
<tr>
<td>Mei Cheng Whale</td>
<td>Course Administrator</td>
<td><a href="mailto:meicheng@cse.unsw.edu.au">meicheng@cse.unsw.edu.au</a></td>
<td>9385-5683 Ext 55683</td>
</tr>
<tr>
<td></td>
<td>Lab tutors</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- For general administrative questions, email en1811@cse.unsw.edu.au

- For info see the [class home page](https://webcms3.cse.unsw.edu.au/ENGG1811/19T2/)
Something about your lecturer
Course Objectives

• What you *should be able to do* by the end of the semester:
  
  – use the Python programming language and its associated packages to solve computational problems
  
  – design and implement solutions to computational problems
  
  – Have a basic understanding of numerical computing environments such as MATLAB® and Microsoft Excel

• **Important**: There are two versions of the Python programming language. We will be using *version 3.6*. 

• The two versions (called Python 3 and Python 2) are slightly different but for compatibility with course materials, lab exercises, assignment marking and exam, always use Python 3.6, **not** Python 2.
Ways of Learning (1)

• Lectures
  – Lecture slides available before the class
  – Code:
    • Before the class: sample code or incomplete examples
    • After the class: Complete examples
  – Ask questions if you wish
  – Only effective if audience cooperates by maintaining silence

• Labs
  – To develop experience with problem solving
  – Supported by tutors
  – Scope: programming exercises + one online multiple choice question
  – Expectations:
    • Complete the work by the end of the lab.
    • Must be ready to show 30 minutes before end
    • Tutors may ask you questions
Ways of Learning (2)

• Assignments
  – Extended programming work that integrates skills from multiple weeks’ of lectures

• Consultation

• Course Forum
  – strongly encouraged to participate
  – usual etiquette:
    • respect for participants' opinions
    • no assignment solutions (tiny fragment is OK to ask a question though)
The proposed lecture schedule is:

<table>
<thead>
<tr>
<th>Week</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weeks 1 to 9</td>
<td>Problem solving and programming using the Python 3 programming language</td>
</tr>
<tr>
<td>Two Virtual Lectures</td>
<td>Matlab and Spreadsheet</td>
</tr>
<tr>
<td>Week 10</td>
<td>Introduction to Machine Learning. Course Revision.</td>
</tr>
</tbody>
</table>
Assessment

- **Labs** (weeks 1 to 10) cover all the topics:
  - **10 in-class labs** during Weeks 1 to 10 in your lab classes.
  - In addition, **two virtual labs**.
  - No marks for Week-01 lab.
  - In-class labs in Weeks 2 to 9 are marked out of 3 marks (2 marks from tutor assessment of your lab work, 1 mark for an on-line multiple choice question).
  - Week-10 lab is marked out of 3 marks.
  - Each virtual lab is marked out of 2 marks. There are two virtual labs.
  - We consider **best 8 in-class lab marks** (24 marks), plus 4 marks for two virtual lab marks. Total lab marks 28.
  - Lab marks contributes **10% of total**.
Assessment

• Assignments
  – *Important: the following due dates are subject to change*
  – Assignment-01 proposed due date on Monday of Week-08 (Python program), 10%
  – Assignment-02 proposed due date on Friday of Week-10 (Python program), 10%

• Mid-Semester Test
  – In Week 6’s lab, test on Python, 10%
  – Scope: Lectures 1-4, Labs 1-5

• Final Exam
  – In computer labs. You will be asked to write computer programs.
  – 60% of overall assessment, minimum competency 33.3% (20/60)

• Mid-term and final: Python documentations available
Avoiding Plagiarism

• Academic honesty
  – everything submitted for assessment must be your own work
  – acknowledge all sources unless obvious

• Assignments 1 and 2
  – program code must be developed alone
  – discussion about solutions OK, indeed encouraged
  – imperfect but honest attempt will still attract fair marks
  – exam-related question carries more weight than the assignment, and will only be solvable if attempted the assignment

• Anti-plagiarism measures
  – start early and get help if you're struggling
  – we usually run sophisticated similarity analysis software
  – mark reduction of up to 100% applies to non-original submissions

• More information in Student Conduct section in the course outline
## Software

<table>
<thead>
<tr>
<th>Software package</th>
<th>When</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spyder (Part of Anaconda for Python 3.6)</td>
<td>Lecture Weeks 1-10, Assignments</td>
<td>Free software</td>
</tr>
<tr>
<td>Matlab</td>
<td>Virtual Lecture / Lab</td>
<td>Available to UNSW students under licensing agreement</td>
</tr>
<tr>
<td>Microsoft Excel</td>
<td>Virtual Lecture / Lab</td>
<td></td>
</tr>
</tbody>
</table>

- All software are available on the CSE lab computers
  - Remote access to lab computers is available

- Getting started section of the course website has written and video instructions on installing Anaconda
Checklist

To start this course off on the right foot, make sure you have done all of the following by the end of week 1.

- **Enrolled** in the course properly (with a lab class)
- Found out **where the labs are**
- Installed Anaconda for Python 3.6 on your own computer
- Had a go at the **first lab** (lab 01)
- Dropped into the course forum, maybe posted a comment

Class home page (yet again):

https://webcms3.cse.unsw.edu.au/ENGG1811/19T2