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/ENGG1811/25T1/

People and Website

Dr. Ashesh Mahidadia

Larry Tang

Staff Name	Role	Email
Dr. Ashesh Mahidadia	Convenor and Lecturer	a.mahidadia@unsw.edu.au
Larry Tang	Course Administrator	en1811@cse.unsw.edu.au
	Lab tutors	

- For general administrative questions, email
  en1811@cse.unsw.edu.au
- For info see the **class home page**
  https://webcms3.cse.unsw.edu.au/ENGG1811/25T1

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 (called Python 3 and Python 2) of the Python programming language. They are slightly different but for compatibility with course materials, lab exercises, assignment marking and exam, **always use Python 3.7** (or ***higher***), **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

- 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
- Online 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)

Assessment

Please note that the following schedule is subject to change.

Item	Topics	Due	Marks	Notes
Labs	All topics	8 labs during Weeks 1 to 10 during your lab classes. In addition, two self-directed labs due in Week 8 and Monday Week 11.	20%	• 8 labs (weeks 2, 3, 4, 5, 7, 8, 9, 10). Each lab is marked out of 3 marks: one mark for an on-line multiple choice question, and two marks for satisfactorily demonstrating your lab work during the respective lab. (16% for 8 labs, 2% each lab). Note your mark out of 3 will be scaled to a mark out of 2 • Each self-directed lab (aka "virtual lab") is marked out of 2 marks. There are two virtual labs. (4% for two self-directed lab)
Assignment 1		Week 7	20%	
Assignment 2		Week 10	20%	
Final Exam	All topics	Exam period	40%	

Calculations of final mark :

You receive the sum of the component marks described above in the table.

Final mark = Labs (20) + Assignment-1 (20) + Assignment-2 (20) + Final exam (40).

You need to get 50 or more marks to pass the course.

Laboratory classes

- 8 labs (weeks 2, 3, 4, 5, 7, 8, 9, 10).
- Each lab is marked out of 3 marks: one mark for an on-line multiple choice question, and two marks for satisfactorily demonstrating your lab work during the respective lab.
- 16% for 8 labs, 2% each lab. Note your mark out of 3 will be scaled to a mark out of 2.
- There are **two virtual** (self-directed) labs. (4% for two self-directed lab).
- Each **virtual** (self-directed) lab is marked out of 2 marks.

Assignments

Two Assignments.

- Assignment-01 :

  available in week 03 or 04, and due in week 07.

- Assignment-02 :

  available in week 07, and due in week 10.

Calculations of final mark

You receive the sum of the component marks described above in the table.

Final mark = Labs (20) + Assignment-1 (20) + Assignment-2 (20) + Final exam (40).

You need to get 50 or more marks to pass the course.

Lecture Schedule

The proposed lecture schedule is:

Week	Topic
Weeks 1 to 9	Problem solving and programming using the Python 3 programming language
Two Virtual Lectures	Matlab and Spreadsheet
Week 10	Introduction to Machine Learning. Course Revision.

Please also refer to the proposed "**Course Schedule**" in the course outline available on the class web page.

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

Software package	When	Note
Spyder [Part of Anaconda for Python 3.7 (or *higher*)]	Lecture Weeks 1-10, Assignments	Free software
Matlab	Virtual Lecture / Lab	Available to UNSW students under licensing agreement
Microsoft Excel	Virtual Lecture / Lab	

- 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 3.7 (or higher) 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/ENGG1811/25T1/