ENGG1811 Computing for Engineers
Semester 1, 2017

Course Introduction

1. Course Aims

ENGG1811 is an introductory computing course that covers the following topics:

- An introduction to popular applications (such as spreadsheets and numerical computing environments) to solve computational problems.
- An introduction to procedural programming (using tools associated with the applications) to solve engineering, business and scientific tasks.
- An introduction to some aspects of current and emerging Information Technologies.

While designed to develop a range of computing skills required for most programs in Engineering, the course is also suitable for other professional disciplines.

2. Student Learning Outcomes

After completing this course, students should be able to

- use spreadsheets and their associated tools to solve small computational problems.
- design and implement solutions by writing small programs with a procedural language such as OpenOffice.org Basic that comply with a given set of requirements.
- use a numerical computing environment such as MATLAB® to analyse, model and visualise data and systems
- understand the limitations and advantages of some of the emerging IT technologies, and how to exploit them in solving engineering problems.

Lab exercises, assignment problems and the midterm test will assess skills you have acquired in the first two topics and three learning outcomes. The final written exam will challenge your understanding in all areas.

3. Teaching, Learning and Assessment

At university, the focus is on the self-directed search for knowledge. Lectures, labs and assignments, with supporting resources, are all provided as a service to assist you in this endeavour. It is your choice as to how much work you do in this course, whether it be preparation for classes, completion of practical work, study for exams or seeking assistance or extra work to extend and clarify your understanding. You must choose the approach that best suits your learning style and goals in this course. Note that the University expects you to do about 180 hours of work for this course including lecture attendance and review, labs, and time spent on assignments. Of course, this will vary according to your aims. The course is designed in such a way that passing the course will require a good understanding of the fundamental notions as well as good practical skills. This can only be achieved through regular and consistent work. If your aim is to obtain a high grade then you will need to invest more time on this course.

Lectures

The aim of the lectures is to facilitate learning and understanding of the important concepts within course syllabus. They will provide the main source of course material delivered. There are three hours of lectures per week.

In the lectures, we will

- explain each topic
- demonstrate the principles underpinning each topic with examples that apply tools (spreadsheets, numerical environments, programming structures) to problems relevant to professionals
- discuss how related problems arise and can be solved

You are invited to ask questions at the end of the class, or at any time if you think the question and answer is relevant to everyone else.

> Discussion with the person next to you in class is *not acceptable*, as it creates a level of noise that inevitably impacts on the rights of others who are trying to listen and learn.

Lecture Schedule

The proposed lecture schedule is:

Week	Topic
Weeks 1 to 3	Limitations of number representation. Spreadsheets: cell addressing, formatting, formulas. Predefined functions, filters, charts. Analysing data: pivot tables, curve fitting, Goal Seek, Solver.
Weeks 4 to 7:	Problem Solving and Programming: Introduction to procedural programming and OpenOffice.org Basic. Data types, variables and constants, operators and statements, selection (If statements) and iteration (While and For statements), modularity using functions and subprograms, parameter passing. Program design, development, testing and debugging. OO's object model. Problem solving case studies using common numerical algorithms.
Weeks 8 to 11	Introduction to numerical computing using MATLAB®. Array and matrix operations, selection and iteration, 2D and 3D plotting. Script and function management and documentation. Problem solving, vectorisation, data sources. Visualising models and applying models to experimental data.
Weeks 11, 12:	Brief overview of some of the current and emerging Information Technologies

Please note that the above schedule is subject to minor change.

Laboratory Classes

All lab classes start in Week 2. Each weekly lab class is 2 hours long.

Laboratory classes give you a chance to **practise** problem solving and programming skills on small, well-defined examples. The examples have been chosen to highlight particular aspects of problem solution, and will give you enough grounding in problem solving to assist you in completing your assignment work. Your tutor will be present in your lab class to answer any questions you may have and to assess your competencies.

The lab exercises will be available during the semester via the course's web page. You must have reviewed the exercises, and preferably attempted at least the first of them, before the class. **Take care not to fall behind in laboratory work, as it is difficult to catch up**.

Surveys consistently rate lab classes as the most valuable learning experience in the course. The lab assessment must be completed within the 2-hour period in the designated week. To avoid tutor overload, you must be ready to have some part of the work ready for assessment 30

minutes before the end of the class. Any work presented after this time may not be able to be assessed by the tutor. We try to design the lab so that the basic concepts are covered in the first part, and this usually carries 50% of your nominal lab marks.

Each lab is marked out of 3 marks: 1 mark from a multiple choice question and 2 marks from your tutor's assessment of your lab work.

During each lab, you need to answer online one multiple choice question from a bank of similar questions, and this carries 1 mark of the nominal 3 marks for each lab. These multiple choice questions will be based on the previous week's lecture material and/or the first part of the present lab exercises. Your tutor will assess your other lab work and provide you with a mark out of 2.

Please note that at the time of marking your lab exercises, your tutor may ask you to solve other similar problems. You need to demonstrate that you can solve lab exercises and related problems, in order to receive any marks for your lab work. In other words, using some else's lab solution is pointless!

If you miss an occasional lab class, it won't matter much as the best 10 lab assessments are used for your lab mark. Similarly, a day or two of illness should not have a serious impact on assignment work if you've started early (as you should). However, if you are absent for an extended period due to illness, then you may apply to the class administrator (email the class account en1811@cse.unsw.edu.au or meicheng@cse.unsw.edu.au) for an extension. Written evidence of the problem will be required. A similar procedure applies if you miss the mid-semester test in week 5 due to a documented medical or similar problem.

You can only have your lab work marked during **your** scheduled **lab class**. It is *not* possible to have your work marked in another class.

Consultation

Each week, starting from week 2, the lecturer will be available to discuss with you any aspect of the course that you are having problems with. This is the most personalised form of teaching for this course, and you should take every advantage of it, particularly if you are experiencing difficulties. When assignments are due many additional consultations will be offered.

Please **check the class web page for current consultation time**s (see "Consultations" in the left panel of the class web page).

Assignments

Assignments give you the chance to practise what you have learned on relatively large problems (compared to the small exercises in the Lab Classes). Assignments are a very important part of this course; therefore, it is essential that you attempt them yourself.

There will be two assignments (subject to change):

- Assignment 1 is due in Week 9 (requirements available in week 6).
- Assignment 2 is due in Week 11 (requirements available in week 9).

Assignments are to be completed in your own time. To maximise the learning benefits from doing assignments, it is essential that you start work on assignments early. Do not leave your assignments until the last minute. If you submit an assignment late, the maximum available mark is reduced by an amount (usually 15%) per day that it is late. Assignments are submitted using a link from the class website.

Assignments 1 and 2: the aim for these assignments is to develop skills in all stages of the programming process: start by understanding the requirements; design a method to solve the problem; refine this design; implement and document the design and test extensively to validate that the program meets the requirements.

Examinations

A 45-minute Mid-Semester Test will be held in Week 5 during your scheduled lab class. It covers the first 3 weeks of lecture material (labs 2 to 4).

The final written examination will be held during the examination period. It will examine all material covered in the course, but will emphasise the material in the second half of the course.

Supplementary Exams

The document "*Essential Advice for CSE Students*" states the supplementary assessment policy for the School of CSE. Please take the time to read it carefully. If you are granted a Supplementary examination, then it will be held on the date specified in the above document. If you think that you may be eligible for a supplementary exam, then make sure you are available on that day. It is your responsibility to check at the School Office for details of Supplementary examinations. Please note that there will be NO further supplementary dates for this course. In other words, we will NOT be able to offer you supplementary examination at any other time.

A student may be considered for the supplementary exam if his/her final marks are in the rage 47 to 49, and the student's performance in the other course components (mid-semester test, assignments and labs) is satisfactory.

Assessment

The assessable components of the course are:

Component	Mark
Assignment 1	8%
Assignment 2	7%
Lab Exercises, best 10 (of 12)	10%
Mid-Semester Test	15%
Final Written Exam	60%

More details about how the topics are assessed are provided in the first lecture (lecture material is available progressively on the class web page)

Important:

- In normal circumstances, you receive the sum of the component marks.
- However, **if your exam mark is poor** (less than one third or 20/60), your final mark is calculated using the formula

  Final = 3.75 * Exam * Other / (Exam + Other)

  where *Exam* is the raw exam mark out of 60 and *Other* is the sum of the other assessments, out of 40. This formula results in a mark that is less than 50.

4. Course Resources: Print and Online Resources

The first two topics have good online documentation, and reference and tutorial sites are linked on the class page. Printed material is the best for the third topic, numerical computing, which begins in week 8. Most engineering students will use Matlab in later courses or to complete senior

projects. Reference 1 is the primary reference, and is probably worth buying for now and later. It is available at the UNSW Bookshop

http://www.bookshop.unsw.edu.au/details.cgi?ITEMNO=9780495668077

References 2 to 4 are also useful alternatives if you happen to have access to a copy.

1. Chapman, SJ (2012). *MATLAB Programming with Applications for Engineers*. Cengage Learning. ISBN 9780495668077.
2. Chapman, SJ (2009). *Essentials of MATLAB Programming*. 2ed. Cengage Learning. ISBN 9780493295686. [an earlier book with almost identical coverage]
3. Moore, H (2012). *MATLAB for Engineers*. 3ed. Pearson. ISBN 9780132103251.
4. Gilat, A (2005). *MATLAB An Introduction with Applications.* 4ed (SI). ISBN 9780470873731.
5. Holloway, JP (2004). *Introduction to Engineering Programming*. Wiley. ISBN 9780471429272. [a C++ programming book but with a good engineering focus]
6. Chapra & Larsen (2013). *Computing for Engineers*. Pearson. ISBN 9781442515208.

The syllabus for ENGG1811 was updated for 2014. In prior years there has been a customised textbook, reference 6, covering spreadsheets (using Microsoft Excel) and programming (using VBA). Although we are using different tools this year, the principles are the same, so if you have access to a copy it may still be beneficial.

We will provide additional resources on the class web page to cover other many useful topics.

5. Administration

Units of Credits

6 units of credit

Pre-requisites

There are no pre-requisites for this course.

Assumed knowledge

We will use mathematics frequently, but mostly informally. Fundamental concepts assumed to be familiar to you include trigonometry, calculus

(functions and derivatives), complex numbers and matrix arithmetic. The latter is covered in MATH1131 in parallel with its use in ENGG1811.

Course Web Page

http://www.cse.unsw.edu.au/~en1811

Staff

Name	Role	Email	Phone (ext)
Ashesh Mahidadia	Lecturer-in-charge	ashesh@cse.unsw.edu.au	prefer email contact
Mei Cheng Whale	Course Administrator	meicheng@cse.unsw.edu.au	93855683 (ext 55683)

Lecture Times

Time	Location
Mon 11am-1pm	Sir John Clancy Auditorium (K-C24-G17)
Wed 11am-12pm	Sir John Clancy Auditorium (K-C24-G17)

Getting Help

From time to time, various problems may arise in your study of this course. Below is a list of typical problems, with suggestions for where you might seek help with them:

Problem	Solution
Help with problems unrelated to this course	CSE Web Help Page
Can't understand lecture material	Ask the lecturer after the lecture times, OR ask your tutor during your lab class, OR go to a consultation.
Stuck with lab work	Ask your tutor during your lab.
Stuck with assignments	Go to a consultation.
Want to change lab class after week 1	Email class account (en1811@cse.unsw.edu.au)

Problems with lab computers	CSE Help Desk
Problems with your account	CSE Help Desk
Problems with Uniwide access etc.	IT Service Centre, Ground floor, Library annexe. 9385-1333.

One final piece of advice. If you have a problem, **do not wait** until late in the semester before you seek help. Problems that are dealt with early are usually easier to resolve than ones that are left to the last minute.

6. Academic Honesty and Plagiarism

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

The pages below describe the policies and procedures in more detail:

- Student Code Policy
- Plagiarism Policy Statement
- Plagiarism Procedure
- Student Misconduct Procedure

All work submitted for assessment **must be your own work**. Lab exercises and assignments **must be completed individually**. We regard copying of assignments or lab exercises, in whole or part, as a serious offence. **We use plagiarism detection software to search for work derived from other students or from other sources.** Be warned that:

- the submission of work derived from another person, or jointly written with someone else will, at the very least, result in zero marks for the assignment or lab exercise.
- allowing another student to copy from you will, at the very least, result in a reduction in the mark for your own assignment or lab exercises; and
- severe or second offences will result in automatic failure of the course, or even exclusion from the University.

Collaborative learning in the form of "think tanking" is encouraged, but students are not allowed to compose programming solutions together as a group during such discussions, unless an assignment explicitly allows this. Students are also warned not to send code fragments of the assignments or labs to each other in any form (e.g. as email or listings). In addition, purchasing code from a solution-generating website or service or from a senior student is a very serious kind of academic misconduct.

7. Continual course improvement

Student feedback on this course, and on the effectiveness of lectures in this course, is obtained via electronic survey (CATEI) at the end of each semester. Student feedback is taken seriously, and continual improvements are made to the course based in part on this feedback. Students are strongly encouraged to let the lecturer in charge know of any problems as soon as they arise. Suggestions and criticisms will be listened to openly, and every action will be taken to correct any issue or improve the students' learning experience.

As a result of recent student feedback and after consultation with all schools in the Faculty of Engineering we replaced the second half of the programming topic with an introduction to MATLAB, which students will find more relevant to their studies. Over recent semesters assignments have also been restructured to allow partial completion for students who find programming particularly challenging.

Document release date: 20-Feb-2017.

Post-release corrections: