

# COMP20005 Engineering Computation

Semester Two, 2016

*Introduce yourself to your neighbours while you are waiting*

July 22, 2016

# Staff

- Coordinator and lecturer: [Jianzhong Qi](#)
- Consultation: [3:30-4:30pm Wednesday, Rm 8.14, Doug McDonell Building](#)
- Head tutor: [Nicholas Brown](#)
- Tutors: [Wenkai Jiang](#), [Ryan O’Kane](#), [Alan Thomas](#), [Anh Vo](#), [Angus White](#), [Regina Zhang](#)
- Demonstrators: [Jonathon Haden Grigg](#), [Yujing Jiang](#), [Gitansh Khirbat](#), [Ce Liang](#), [Dongge Liu](#), [Tian Luan](#), [Wenxi Wang](#), [Yiqing Zhang](#)
- Contact information on LMS “Staff Information” page

# Students

- Primarily intending engineers, who need to know something about programming; plus some mathematicians, physicists, etc.
- The single most important thing you have to do to get the semester off to a good start is to **make sure you have friends in the class.**
- So, every time you enter a room over the next two weeks, sit down next to someone you don't know, and introduce yourself.

## Critical information

- All of the key information is provided in the introductory handout.
- All handouts and announcements will also be provided via the LMS page. You should check it every 2-3 days.
- Lecture recordings will appear on the LMS page shortly after each class.
- Workshops will commence in [Week Two](#).

# Workshop time table

Day	Time	Reg.	Building	Room
Thu.	17:15	16	162 (Alice Hoy)	211
Thu.	16:15	24	162 (Alice Hoy)	222
Wed.	15:15	25	162 (Alice Hoy)	236
Wed.	17:15	18	162 (Alice Hoy)	108
Tue.	17:15	20	162 (Alice Hoy)	222
(Canceled) Tue.	17:15	1	162 (Alice Hoy)	108
Tue.	12:00	25	162 (Alice Hoy)	222
Mon.	17:15	18	162 (Alice Hoy)	211
Tue.	16:15	25	162 (Alice Hoy)	211
(Canceled) Mon.	19:15	2	162 (Alice Hoy)	211
Tue.	12:00	25	162 (Alice Hoy)	236
Tue.	18:15	5	162 (Alice Hoy)	211
(Canceled) Thu.	19:15	3	162 (Alice Hoy)	236
(Canceled) Thu.	18:15	0	162 (Alice Hoy)	222
(Canceled) Tue.	17:15	0	162 (Alice Hoy)	109

# Subject overview

- Engineering Computation provides an introduction to programming and algorithms using the language C.
- You can use any C programming environment that you have access to. The Engineering labs support two different mechanisms, one based on [jEdit](#) and command-line compilation, and one based on the [Eclipse](#) integrated development environment.
- Both approaches are free and can be installed on home computers and laptops; [jEdit](#) is the simpler of the two.

# Subject overview

- The emphasis is on **you** doing programming, and learning the necessary skills in a **hands on** manner. We will be making use of numeric problems, and, where possible, engineering examples.
- You need to work steadily through the semester, and write (and execute) programs throughout.
- Programming is like riding a bike, you need lots of actual practice to become good at it.

## Overlap with COMP10002 Foundations of Algorithms

- Both COMP10002 and COMP20005 include C programming as a core component (and have the same textbook).
- In Foundations of Algorithms, the emphasis is on data structures and algorithms.
- In Engineering Computation, the emphasis is on using basic C language to solve numerical problems.
- Enrolment in both should be avoided. Seek advice if you are uncertain.



# Class and attendance

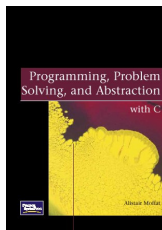
- There are three lectures each week, plus a two-hour workshop.
- Each workshop has two parts: a tutorial space used for initial discussion, and a laboratory space used for individual programming.
- You should attend **all** scheduled classes.

# Computer accounts

- You may use any of the School of Engineering computer labs in Old Engineering and in the Alice Hoy Building (when free of scheduled classes).
- Your standard University account name and password will allow access.
- Your University email address (something like [jsmith@student.unimelb.edu.au](mailto:jsmith@student.unimelb.edu.au)) should be directed to a email address that you read regularly.

# Textbook

- The prescribed text is **Programming, Problem Solving, and Abstraction with C** by Alistair Moffat (2013 revision, <http://people.eng.unimelb.edu.au/ammoffat/ppsaa/>).
- It is on sale at the Co-op Bookshop. There is also an e-edition available from the publisher's website.
- Second-hand copies of the yellow first-edition are also available and may be used.



# Total workload

- Three lectures, and a two-hour workshop. Plus:
  - A review hour per lecture, including reading the text.
  - Two preparation hours for the workshop.
  - Two hours of general review/reading, perhaps in a study group.
- In total, around 12 hours per week per subject is required, starting immediately.
- Make a study timetable for all activities, **and stick to it.**

## Total workload

- If you have outside interests (including work) that consume more than approximately 12-15 hours per week, you are seriously jeopardizing your chances of passing.
- If your outside interests cannot be restricted to fewer than 12 hours per week, you should consider taking only three subjects per semester.

## Seeking assistance

There are a range of mechanisms to use when you need help (in order of priority):

- Check the LMS for general announcements.
- Post your query to the LMS discussion forum. Read other posts and responses while you wait for a response to your query.
- Ask a question after a lecture (or before, if you believe the answer will be of wide interest).
- Consultation: 3:30-4:30pm Wednesday, Rm 8.14, Doug McDonell Building
- Make an appointment (email the lecturer; [start email subject with COMP20005](#)).

# Assessment

Your final mark is the combination of three components.

- Mid-semester test (Week 6, tentative), 10%
- Workshop projects (Week 8 and Week 12, tentative), 30%
- Examination, 60%

To pass the subject as a whole you must attain at least **28/70** (combined) in the test and exam; plus **12/30** (combined) in the two projects; plus **50/100** overall.

## Mid-semester test

- The test will take place in Week 6 in the usual class time. You should use the test as early feedback of your status in this subject.
- If you do well, that's great.
- If you do poorly, heed the signal it sends.
- A sample test, and more details of the format, will be supplied closer to the time.



# Academic honesty

- All assessed work in this subject is **individual**.
- It is easy for us to run sophisticated similarity checking software over all submissions.
- The University's Academic Honesty policy will be applied if duplicate work is detected.

# Academic honesty

- There are also rules governing misuse of the various computer systems.
- Misuse includes unauthorized storage of copyright material (software as well as digital data like music); unauthorized access to other accounts; and any other activity not associated with your study.
- Choose a sensible password, and keep it secure.

## Week One checklist

- Check that you can access the LMS page.
- Confirm your workshop time, and check the LMS for any late messages about workshop locations.
- Buy the textbook. Read Chapters 1 and 2.
- Most importantly, make some new friends, have some fun, and get set for a great semester.

## When something goes wrong. . .

- This is a big subject, and there are bound to be some things that go wrong. Nor can we fix things we don't know about.
- So please send email if you can see an issue brewing.
- Hardware faults are notified using the IT-Help page.

## Feedback from last offering

- Provide more sample solutions to textbook exercises
- Encourage working on lecture exercises and textbook exercises
- Provide compulsory homework
- Encourage use of Discussion Forum on LMS
- **Action: allow anonymous posts in Discussion forum; create a sub-forum for posting textbook exercise and lecture exercise questions and sample solutions.**
- Give a quick review in first hour of workshops
- **Action: now compulsory for the tutors to do that**
- Improve assignment submission system

# The first C programs

- `helloworld.c`
- `addnumbers.c`