

## **Department of Computing and Information Systems**

### **COMP20005 Engineering Computation**

**Semester 2, 2016**

#### **Subject description**

COMP20005 Engineering Computation is designed to give Engineering students the programming skills and confidence to solve numeric problems using a computer. It presents fundamental principles using the programming language C, with an emphasis on the computer system from a programmer's viewpoint rather than from a computer designer's viewpoint. A range of algorithmic techniques for solving common numerical and engineering problems are also presented.

While COMP20005 is not restricted against COMP10002 Foundations of Algorithms, both include coverage of C programming, and there is non-trivial overlap between them. Students who have already completed COMP10002 should seek course advice from a staff member.

#### **Staff**

The following people are involved in this subject in Semester 2, 2016:

Coordinator and lecturer: Dr. Jianzhong Qi

- [jianzhong.qi@unimelb.edu.au](mailto:jianzhong.qi@unimelb.edu.au)
- Room 8.14 Doug McDonell Building (Building 168)

Head Tutor: Nicholas Brown

- [nicholasb1@student.unimelb.edu.au](mailto:nicholasb1@student.unimelb.edu.au)

Your Tutor: 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

The coordinator is responsible for the overall organization of the subject and for the selection of lecture and workshop material. Assistance is provided by the tutors who manage the workshops, all of who are senior/former students of the Department of Computing and Information Systems employed on a casual basis. Make a note of the name of your tutor at your first workshop.

A staff-student meeting will be held during the semester to allow feedback. Nominations for representatives will be called for early in the semester.

#### **Lectures**

Each week, you are expected to attend three lectures. The purposes of lectures are:

- To present the principles of the subject, illustrated by examples;
- To introduce the material that will be developed in workshops;
- To distribute any printed material required for the subject; and
- To make announcements about the subject, particularly about the syllabus and assessment.

Much of the information presented in lectures is available from other sources, primarily the text book, and the LMS page at <http://app.lms.unimelb.edu.au/>. Lecture recordings are also available via the LMS. Even so, you should endeavour to attend all of your scheduled lectures.

## Workshops

Each week *starting from Week Two of semester*, you will be expected to attend a two-hour workshop. In the preceding week, you will be given a set of exercises to reinforce and extend the lecture material and to help consolidate your understanding. The main purposes of workshops are:

- To give you an opportunity to raise questions about the subject;
- To clarify any problems you are having;
- To discuss alternative solutions to the assigned exercises;
- To independently implement one or more of those solutions; and
- To develop your confidence in using computers to solve numerical problems.

The rooms allocated for the workshops reside in different buildings. Classes will usually commence with a tutorial, so that the tutor can introduce the week's work, and so that students can readily discuss their proposed solutions. At some point within the two-hour period the group will then move on to working individually on their solutions.

You will get much more benefit from the workshops if you ask questions and contribute to the initial discussion period. *That can only happen if you prepare in advance. It is important that you work through the assigned exercises before attending your assigned class.* If you do not adequately prepare, you will probably not be able to finish all the work during your lab time. Although School of Engineering computer laboratories may be available (more likely to have been fully booked already) at times outside of your scheduled class, you will find it much easier to complete your work when you have a tutor available to assist you with problems.

## Expectations

The LMS page (menu item "Expectations") gives detailed guidance on what is expected of you (and of staff) in lectures and workshops.

## Total workload

In total, we expect you to spend about 10–12 hours per week on this subject, starting in Week One and continuing through until the exam. To help you reach this target, you are encouraged to prepare a complete study timetable that shows, for this subject, the following 12 hours:

- Three hours of lecture attendance, plus three hours of follow-up lecture review. In each review hour you should work through the content of the previous lecture to consolidate your understanding, and read the relevant sections of the textbook. (Six hours in total).
- Two hour of workshop, plus two hours of preparation prior to that class. (Four hours in total).
- Two further hours of review and/or reading, perhaps including a Study Group meeting with other students enrolled in this subject, and/or working independently on the project work.

When the similar demands from your other three subjects are fitted into the equation, it is clear that you need to spend 40 to 48 hours per week on your University study. *If you have outside interests (including work) that consume more than approximately 12–15 hours per week, you are seriously jeopardizing your chances of successfully completing a full-time subject load. 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

The LMS should be your first port of call when seeking help. Issues that affect multiple students are likely to have been picked up and answered via the “Announcements” page. More specific questions can be posted via the LMS Discussion Board for staff – or other students – to answer. Even if you don’t have questions of your own, you are likely to be able to benefit by reading other students’ queries, and will perhaps even be able to post answers to help them out. Staff will routinely monitor the LMS forum, and when necessary, provide additional answers to questions.

You should also feel free to approach the lecturer. Immediately after each lecture is usually a good time to ask quick questions, or to request appointments for longer discussions. You may also make contact by email to ask questions and to set appointments; the LMS “Staff Information” page will be kept up to date with any scheduled office hours that might be of interest.

## Assessment

A mid-semester test will be held in the lecture on Week 6 (tentative) worth 10% of your final mark. Details of the mid-semester test will be made available closer to the date of the test. To help you prepare, a sample test will be provided. There will be test viewing sessions arranged.

There will also be regular project work, to be completed during (and then after) the workshops, including two assessed programming projects that account in total for 30% of your final grade. The two projects will be due at Week 8 (tentative) and Week 12 (tentative).

The written two-hour examination at the end of semester is worth 60% of your overall mark. The exam will require detailed knowledge of the workshop exercises and projects, so it is important that you understand *all* of the programming work.

To pass the subject as a whole, you must obtain at least 50% overall when all marks are combined; must obtain at least 12/30 in the project work; and must obtain at least 28/70 in the mid-semester test and end-of-semester written examination when those two marks are combined.

## Academic honesty

In this subject all assessed work is to be completed on an individual basis. You should be aware that we have access to sophisticated similarity checking software to automatically identify pairs of programs that have similar sections, even when variable names have been altered. If duplicate submissions are detected, both parties – receiver and giver – will be referred to the School of Engineering for handling under the University Discipline procedures. For further information, see <http://academichonesty.unimelb.edu.au>.

The University also has penalties that apply for misuse of computing facilities. You should respect copyright, and not store any unauthorized copyright material on any University computer; should refrain from accessing accounts and files other than your own; should not use University computing facilities for any non-study purposes; and should keep your own account secure.

## Texts

The prescribed text is *Programming, Problem Solving, and Abstraction with C* by Alistair Moffat (revised edition, Pearson, 2012, with the blue cover). You will need to consult this text on a regular basis, and should probably have your own copy. Information related to the text (including an errata listing) is at <http://people.eng.unimelb.edu.au/ammoffat/ppsaa/>.

Second-hand copies of the first edition (with the yellow cover) can also be used. Note that the second edition is cheaper than the first edition was; the Co-op Bookshop has a member price of \$70. An e-edition is available from the publisher's website for \$49.

There are also other texts on programming and C which you might find helpful as reference. One such book is *The C Programming Language*, by Kernighan and Ritchie (Prentice-Hall).

### **Subject web page**

All subject information, such as the exercises set for the weekly workshops, are posted on the subject LMS page at <http://app.lms.unimelb.edu.au/>, as are all handouts (including this one). You are expected to visit the LMS every two or three days and read any subject announcements.

### **Computer laboratories**

All computing in COMP20005 will be done in the Melbourne School of Engineering (MSE) computer laboratories. All of the MSE laboratories may be used when not occupied for scheduled lab classes. Your University account name and password are used to access the computers.

You will have had a university email account created as part of your enrolment, something like [jsmith@student.unimelb.edu.au](mailto:jsmith@student.unimelb.edu.au), and should ensure that you read it regularly.

### **Working at home**

The software tools used in this subject are available as free downloads, and if you install them on your home computer or laptop, you will be able to work away from the MSE laboratories. Details of how to do this are linked from the LMS page. Note that in the end all submitted work must reside in the file systems attached to the lab computers so that it can be viewed by us.

You are also permitted to use any other C programming environment available to you, including the gcc compiler that operates under cygwin on PCs, and under MacOS on a Mac. Another useful tool is jEdit, see <http://www.jedit.org/>. Note that knowledge of the operation of any particular software development tools is NOT part of the examinable content of this subject, jEdit and other similar editing environments are purely tools that facilitate your learning of C programming.

### **To maximize your chances of success...**

The most important thing you can do through the semester is to step methodically through the assigned exercises, making them work on the computer and then exploring variations, doing your own “what if” experimentation. Doing this with another person will be even more rewarding, and having a broad group of friends taking this subject is the second important thing. A strong support network means that you are much less likely to get left behind by lecture material, and regular discussions of the subject content as part of a study group – both asking and answering questions – is also beneficial. Early in the semester is always a good time to make new friendships, so for the next few weeks don't hesitate to introduce yourself to others at the start of each lecture and workshop.

### **Welcome to Engineering Computation, 2016!**

*Drafted by Alistair Moffat and edited by Jianzhong Qi,*

*Department of Computing and Information Systems,*

*July 2016*