School of Computing and Information Systems

comp10002 Foundations of Algorithms
Semester 2, 2019

Subject description

The subject comp10002 Foundations of Algorithms is designed to reinforce your programming skills and computing knowledge. In it, you will learn the language C, and how C programs are structured, compiled, and managed; and gain an "under the hood" perspective of computers and programming. At the same time, the subject is about algorithms, the fundamental building blocks of computation. Topics include dynamic data structures, and the algorithms that manipulate them (lists, trees, hash tables); searching algorithms including pattern searching; and sorting algorithms. More generally, you will be expected to develop your own ability to synthesize algorithmic approaches, and to evaluate problems and computational tasks in terms of what algorithmic techniques might be applicable.

Both comp10002 and comp20005 Engineering Computation include coverage of C programming, and are restricted against each other. Students who have already completed comp20005 Engineering Computation and wish to take a further computing subject should enrol in comp20003 Algorithms and Data Structures. Seek advice if you are unsure.

Staff

The following people are involved in this subject in semester 2, 2019:

	Name	Email	Room
Lecturer	Professor Alistair Moffat	`ammoffat@unimelb.edu.au`	DMD 9.26
Lecturer	Dr Artem Polyvyanyy	`artem.polyvyanyy@unimelb.edu.au`	DMD 10.02
Your tutor			—

The lecturers are responsible for the overall organization of the subject and for the selection of lecture and workshop material. The tutors manage the workshop classes and are senior students in the Department of Computing and Information Systems employed on a casual basis. Make a note at your first workshop class of the name of your tutor.

A Staff-Student meeting will be held during the semester to allow feedback. Nominations for a representative will be called for early in the semester.

Lectures

Three lectures will be provided each week. 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 endeavor to attend lectures; the live experience takes more effort, but is also superior to the recorded experience. On the other hand, if you don't come to lectures, you are more likely to fall behind; should that happen, you will also be less likely to have friends in the subject to help you study.

Because of the large enrolment in the subject this semester, many of you have (only) been able to register for a on-line version of the lectures. Nevertheless, assuming that lecture attendance rates follow the usual pattern, we believe that everyone who wants to attend any given lecture will be able to do so. No-one should excluded from the lectures.

Workshops

Each week, *starting in Week Two of semester*, you will be expected to attend a two-hour workshop. 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 problems.

You will get much more benefit from the Workshops if you ask questions and contribute during 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 may not be able to finish all the work during your lab time, and it is much easier to learn when you have a tutor available to answer questions.

Note that all workshops scheduled for Alice Hoy 210 and 222 will be "bring your own device" classes. If you have registered for a class in one of these two rooms and are not able to bring your own computer to class each week you should switch to another class. There will be technical support provided in the first classes in those two rooms to help you install the required software.

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, you need 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 twelve hours:

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

When the similar demands from your other subjects are fitted into the equation, it is clear that you need to spend 40–48 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 answered via the "Announcements" page. More specific questions can be posed 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 benefit by reading other students' queries, and will perhaps be able to post answers to help them out. Staff will routinely monitor LMS forums, and when necessary, provide additional answers to questions.

You should also feel free to approach the lecturers. Straight after lectures is normally the best time to ask questions. You may also make contact by email to ask questions and seek advice.

Assessment

There will be a mid-semester test held during the scheduled lecture time on Monday 2 September worth 10% of your final mark.

There will also be regular project work, to be completed during (and then after) the workshops, including two assessed programming projects (due September 23 and October 21) that account in total for 30% of your final grade.

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. We make use of sophisticated similarity checking software to automatically identify pairs of programs that have similar sections, even when variable names have been altered and other "deceptions" put in place. 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://academicintegrity.unimelb.edu.au`. In the past such referrals have led to allegations of Academic Misconduct being upheld, with penalties that have included students being awarded zero for the subject, regardless of marks received for other assessment components. Nor may you request others to complete your assessed programming work, whether or not payment is involved. Submission of work authored by others may lead to termination of enrolment. You will be required to submit an "Authorship Declaration" with both assignments, and will incur a substantial mark penalty if you do not do so.

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 purposes other than those connected with your study; and should take responsibility for keeping your own account secure.

Texts

The prescribed text is *Programming, Problem Solving, and Abstraction with C* by Alistair Moffat (revised edition, Pearson, 2012). You will be expected 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/`. The Co-op Bookshop has a member price of $\$75$, and an e-edition is available from the publisher's website for around $\$55$. There are many second-hand copies for sale.

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

Subject web page

All subject information will be posted on the subject LMS page at `http://app.lms.unimelb.edu.au/`. You are expected to visit the LMS every two or three days and read all subject announcements. Lecture recordings are also available on the LMS page. But note, watching the lecture second-hand is **never** the same as watching it live. Please don't make the mistake of thinking that it doesn't make a difference if you don't attend classes. It most certainly **does** make a difference.

Computer laboratories

All computing in comp10002 will be done in the Melbourne School of Engineering computer laboratories in the Old Engineering Building, Bouverie Street Building, Electrical Engineering Building, and in the Alice Hoy Building. All of these laboratories are open for casual use when not occupied for scheduled classes. Your standard University account name and password are used to access the computers. You will have a university email account as part of your enrollment, something like `jsmith@student.unimelb.edu.au`, and should ensure (by forwarding it to another address if necessary) 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 MSE laboratories. Details of how to do this are linked from the LMS page. The source files (with `.c` extensions) can be transferred between home and lab on a memory stick; but 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 that is available to you, including the `gcc` compiler that operates under the `cygwin` suite of tools on PCs, and under MacOS on a Mac. Another useful tool that you might wish to explore 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, and that `jEdit` and other similar editing environments are purely tools that facilitate your learning of C programming.

To maximize your chances of success...

The single 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 you need. 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 Foundations of Algorithms, 2019!

Alistair Moffat and Artem Polyvyanyy,
School of Computing and Information Systems,
July 2019.