COMP1911 Course Introduction

Aims

The objective of this course is for students to develop proficiency in programming in a high level imperative language and to develop a background of relevant knowledge and skills on which to base further study of computing.

Topics covered include:

- fundamental programming concepts
- program testing and debugging
- the underlying memory representation of data
- unix literacy
- version control
- programming style
- professional issues

Practical experience of these topics is supplied by laboratory programming exercises and assignments.

After completing this course, students should be able to design, construct, document and test C programs that accurately comply with a given specification. In addition, the student should also have an understanding and use of common data structures to solve problems. Your understanding should also include knowledge of how programs translate from C into assembler; and how such programs are executed.

Prerequisites

There are no pre-requisites for this course.

Course Web Site

http://www.cse.unsw.edu.au/~cs1911

Staff

Staff Name	Role	Email	Extension
Alan Blair	Lecturer-In-Charge	blair "at" cse.unsw.edu.au	9385-7131

Reading Materials

The recommended text for this course is:

> Alistair Moffat, *Programming, Problem Solving, and Abstraction with C*, Pearson Educational, Australia, 2003, ISBN 1-74103-080-3.

The following books might also serve as additional reference material:

> Brian W. Kernighan and Dennis M. Ritchie, *The C Programming Language*, 2nd edition, Prentice Hall, USA, 1988, ISBN 0-13-110370-9.
>
> Jeri R. Hanly and Elliot B. Koffman, *Problem Solving and Program Design in C, 4th edition*, Addison Wesley, 2004, ISBN 0-321-21055-7.
>
> H.M. Deitel, P.J. Deitel, *C How To Program*, 4th edition, Prentice Hall, New Jersey, 2004, ISBN 0-13-122543-X
>
> A.S. Tanenbaum, *Structured Computer Organisation*, 4th edition, Prentice Hall, New Jersey, 1999. ISBN 0-13-0204358.

Lectures

These are the provisional lecture times and locations:

Time	Location
Mon 11-12	Rex Vowels Theatre
Tue 2-3pm	Law Theatre G04
Thu 2-3pm	Science Theatre

Tutorials

Tutorials give you a chance to clarify ideas mentioned in lectures and to practice your problem-solving skills in a small (and hopefully more personal) class with the assistance of a tutor. You should make sure that you use them effectively by examining in advance the material to be covered in each week's tutorial, by asking questions, by offering suggestions and by generally participating. The tutorial questions will be posted on the Web in the week before each tute. Tutorials are one hour long and they are scheduled to start in Week 2 of session. There will often be a couple of extra questions, which are not covered during the tutorial. You may use these for practice or revision.

Students may also be required to make a brief presentation, in pairs, at the beginning of one tutorial during the session.

Laboratory Classes

Laboratory Classes give you a chance to **practice** programming skills on small, simple examples. The examples have been chosen to highlight particular aspects of programming, and will give you enough grounding in programming practice to enable you to complete your project work relatively easily. Your tutor will be present in your Lab Class to answer any questions you may have.

All lab class exercises must be shown to your tutor during your scheduled lab class. The lab exercises will be available during the session via the course's Web page. **Students are advised not to fall behind in their laboratory work**. In order to get a mark for a lab exercise you must show your tutor a completed or partially completed solution before the end of your scheduled lab class in the week that the exercise is scheduled. If you cannot complete the exercise by the end of the lab you may complete it in your own time and submit it using the `give` command by midnight on the following Sunday. Your tutor will then mark it at the very beginning of the next scheduled lab class. **Note:** To gain a mark you must at the very least: show your tutor a partial solution; and, submit a completed solution using `give` by midnight the following Sunday. **After that time, you will not receive a mark for that particular lab exercise**. If you are unable to attend your scheduled lab class due to illness or misadventure, then you should apply for special consideration and an extension may be granted. **Note:** 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.

To make sure that you can complete the work in your scheduled lab class, you should make an effort to prepare the work beforehand. You can make use of the Lab simply by booking a time-slot. You have sufficient bookable Lab time available each week to allow you to prepare your lab exercises; make good use of it.

Your weekly lab class is 2 hours long and it starts in Week 2 of session. Where possible, your lab class will be scheduled to follow on from your weekly tutorial and will have the same tutor (but you will need to walk to a new room).

Assignments

There will be three programming assignments. Assignments give you the chance to practice what you have learnt 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.

Assessment

The assessable components of the course are:

Component	Mark
Tutorial Presentation	1%
Lab Exercises	9%
Assignments	30%
Written Exam	60%

All submitted programs (lab exercises and assignments) must adhere to the Course Style Guide.

Important:

- Students with a mark less than 10/40 for the practical component (assignments, lab exercises and tutorial presentation) will not be allowed to attempt the final written exam.

Supplementary Exams

The document "Important Advice for 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 is most likely to be held in December. If you think that you may be eligible for a supplementary exam, then make sure you are available at that time. It is your responsibility to check at the School Office for details of Supplementary Examinations.

Plagiarism

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 very serious offence. **We use plagiarism detection software to search for suspiciously similar submissions.** Collaborative work in the form of "think tanking" is encouraged, but students are not allowed to derive code together as a group during such discussions. 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, copying/purchasing of code that is available on the Web is also not permitted. Students who are singled out during our regular plagiarism sweep will be dealt with according to School Policy, which provides serious penalties particularly in the case of repeat offenses:

> http://www.cse.unsw.edu.au/~studentoffice/policies/yellowform.html
> http://www.cse.unsw.edu.au/help/doc/primer/node42.html
> http://www.cse.unsw.edu.au/~chak/plagiarism

Students are strongly advised to protect their work. Do not leave your terminal/computer unattended, or leave listings at the printer for others to take.

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 your tutor during your tute/lab
Stuck with assignments, lab work	Ask your tutor during your tute/lab
Problems with lab workstations	CSE Help Desk
Problems with your account	CSE Help Desk
Problem with Dial-Up access from home	DIS-Connect Helpdesk (9385 1777)

The School Web site has useful information that may be of help:

> http://www.cse.unsw.edu.au/

Particularly useful is the Unix `man` pages. The searchable interface allows you to find out more about various Unix commands (this assumes that you have some basic knowledge of the Unix environment):

> http://www.cse.unsw.edu.au/scripts/man

If you are new to the computing laboratory and its setup, you can read about it in the *Unix Primer*. The *Unix Primer* and other useful documentation can be found at this link:

> http://www.cse.unsw.edu.au/help/doc/index.html

Take the time to explore the system. The more adventurous you are, the more you will learn about the system and the more proficient you will get. Those interested in working from home should look into purchasing a copy of our *Home Computing CD*, which is available from the CSG Help Desk. It contains a collection of programs and utilities used by various courses within the School. To find out more, go to:

> http://www.cse.unsw.edu.au/~homecomputing

One final piece of advice, if you have a problem, do not wait until Week 14 before you seek help. Problems that are dealt with early are usually quicker and easier to resolve than if they are left to the last minute.

Course Feedback and Improvement

Student feedback on this course will be obtained via electronic survey at the end of session, and will be used to make continual improvements to the course. Students are also encouraged to provide informal feedback during the session, and to let the lecturer in charge know of any problems, as soon as they arise. Suggestions will be listened to very openly, positively, constructively and thankfully, and every reasonable effort will be made to address them.