Course Outline

COMP1917 Higher Computing 1
Teaching Period 2, 2009

Contents

- Staff
- Course Details
- Aims
- Student Learning Outcomes
- Assumed Knowledge and Relation to Other Courses
- Teaching Rationale and Strategies
- Assessment
- Academic Honesty and Plagiarism
- Course Schedule
- Resources for Students
- Course Evaluation and Development

Staff

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

Consultation Times: TBA

Course Details

Course Code/Title: COMP1917 Higher Computing 1
Units of Credit: 6
Course WebSite: http://www.cse.unsw.edu.au/~cs1917/09s2
Handbook Entry: www.handbook.unsw.edu.au/undergraduate/courses/2009/COMP1917.html

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
- programming style

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

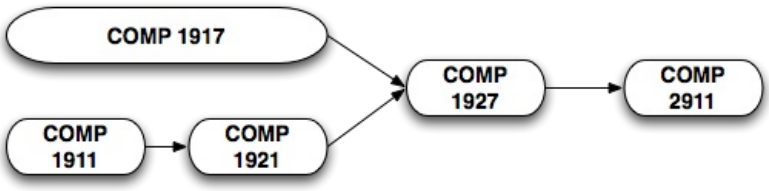
Student Learning Outcomes

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, students should have an understanding of how to use common data structures to solve problems, and how programs are translated from C into machine language and executed.

Assumed Knowledge and Relation to Other Courses

There are no prerequisites for COMP1917.

COMP1917 is designed for CSE students, and for any student with a keen interest in computing, regardless of their degree program. COMP1911 and COMP1921 cover similar material to COMP1917 and (part of) COMP1927, but move at a more gentle pace.



Switching between COMP1911 and COMP1917 is possible up to the end of Week 2. Neither of them assume any prior knowledge of computing, so you should make the choice based on your level of interest and aptitude, rather than on previous programming experience. If you are thinking of switching, the best advice is to look through the course notes, check out the material on the course Web sites and, if necessary, discuss with the Lecturers, the School office or your Program Coordinator.

Teaching Rationale and Strategies

Students are required to attend 3 hours of Lecture per week, plus a 1-hour Tutorial and a 2-hour Laboratory class.

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 the material to be covered before each week's tutorial, asking questions, offering suggestions and 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 will also be required to make a brief presentation, in pairs, at the beginning of one tutorial during the session.

Laboratory classes give you a chance to practice programming skills on small examples, which have been chosen to highlight particular aspects of programming. Your tutor will be present in your Lab class to answer any questions you may have.

All Lab 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 a terminal in a laboratory 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.

There is no scheduled Lab class in Week 1. However, there is a Laboratory exercise for you to complete in your own time during Week 1 (or at least, prior to your Week 2 Lab class).

Assessment

The assessable components of the course are:

Component	Mark
Programming Assignments	30%
Laboratory Exercises	9%
Tutorial Presentation	1%
Practical Exam	30%
Written Exam	30%

All submitted programs (Lab exercises and Assignments) must adhere to the Course Style Guide. 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.

To pass the course, you must achieve:

- at least 15/40 for (Assigments + Labs + Presentation)
- at least 12/30 for the Practical Exam
- at least 12/30 for the Written Exam
- at least 50/100 overall

The document "Important Advice for Students" states the supplementary assessment policy for the School of CSE. Please take the time to read it carefully. COMP1917 students with a mark less that 15/40 for (Assignments + Labs + Presentation) will NOT be eligible for supplementary assessment.

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.

Academic Honesty and 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 offences:

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

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

Course Schedule

These are the provisional lecture times and locations:

Time	Location
Mon 11-12	Central Lecture Block 6
Wed 9-10	Central Lecture Block 6
Wed 2-3	Central Lecture Block 6

Here is the list of planned topics, an estimate of the week in which the topic will be presented, and the relevant chapter/section of the textbook.

Topic	Week	Chapter/Section of Moffat textbook
1. Introduction	1	Chapter 1
2. Numbers In, Numbers Out	1	Chapter 2
3. Making Choices	2	Chapter 3
4. Loops	2	Chapter 4
5. Functions	3	Chapter 5
6. Binary and Hexadecimal	4	Section 13.2
7. Number Storage and Accuracy	4	Section 13.2
8. Characters and Arrays	5	Section 7.1-7.5
9. Pointers	6	Chapter 6
10.Strings and Files	7	Section 7.6-7.10, Chapter 11
11.Writing a Makefile	8	
12.Debugging	8	
13.Structures	9	Chapter 8
14.Linked Lists	9	Section 10.1-10.2
15.Stacks and Queues	10	Section 10.1-10.2
16.Memory and Stack Frames	10	
17.Machine Language	11	
18.Sorting and Efficiency	11	Section 12.1,12.6
19.Review	12	

Resources for Students

For your convenience, most of the lecture notes for this course have been compiled into a volume of Course Notes, available from the UNSW Bookshop:

> COMP1917 Higher Computing 1: Course Notes

These Course Notes do not necessarily cover the entire content of the course, and are meant to be used in conjunction with regular lecture attendance, tutorial and lab attendance, lab exercises, assignment work and study of the relevant sections of the textbook.

The recommended textbook 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.

Since for many of you this is your first time studying computing at university, you are likely to have quite a few questions about a variety of things. If you do have a question please follow the chain of enquiry below to ensure a timely response:

1. check the subject Message Board or CSE forums for an answer to your question
2. post a question on the subject Message Board or CSE forums
3. ask one of your COMP1917 peers
4. ask your tutor
5. attend a consultation session

For problems with Lab workstations or your account, you should contact the CSE Help Desk.

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

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

Particularly useful are 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 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

Specific information for Mac users can be found here:

http://www.cse.unsw.edu.au/~cs1917/09s2/mac.html

One final piece of advice: if you have a problem, do not wait until Week 12 before seeking help. Problems that are dealt with early are usually quicker and easier to resolve than if they are left to the last minute.

Course Evaluation and Development

This course was created in response to student feedback from the course COMP1911 in 2006/07. When it ran for the first time in Session 1, 2008, students expressed satisfaction with the course, but felt that the workload was a bit too heavy. We have therefore made changes to the course, and we believe that the workload is now appropriate. In response to feedback from 2008 and Session 1, 2009, the 3-hour written exam has been replaced by a 2-hour practical exam plus a 2-hour written exam.

Student feedback for 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 received constructively, and every reasonable effort will be made to address them.

Good Luck!
