COMP1911 - Computing 1A - Course Outline

Course Staff

Staff Name	Role	Email	Extension
Andrew Taylor	Lecturer & Admin	andrewt@cse.unsw.edu.au	55525

Lecture Times

Day	Time	Room
Monday	11:00-13:00	Mathews Theatre A
Wednesday	11:00-13:00	Mathews Theatre A

Communication

Sometimes urgent information may be sent to you by email. Make sure you pay careful attention to any email you receive.

All official email will be sent to your CSE email address. If you prefer to read your mail at some other address you can redirect your mail using mlalias. Ask your tutor or the help desk, if you need help doing this. Make sure any forwarding works.

Additional information will be provided in the Course Forums (linked to the class home page). You should read these regularly. These forums are the best place to ask questions about the course.

Pre-requisites

COMP1911 has no prerequisites and assumes no background knowledge.

No previous programming experience is assumed. If you have previous programming experience you should consider taking COMP1917 instead. **Course Details**: 6 UoC, no parallel teaching

Course Aims

This course is designed to introduce students to the basics of programming. It has no prequisites and assumes no previous programming or other knowledge.

Topics covered include:

- fundamental programming concepts
- the C programming language & use of a C compiler
- programming style
- program design and organisation concepts
- program testing and debugging writing programs to solve problems

Course schedule

Week 1	Course Intro, Intro to Computers/Programming, Constants, Variables, printf
Week 2	scanf, expressions, if then else
Week 3	Loops,
Week 4	Arrays
Week 5	Sorting & Searching
Week 6	Characters & Strings
Week 7	Functions
	Mid Semester Break
Week 8	
	Assignment 1 due
Week 9	Pointers, Command Line Arguments
Week 10	File Operations
Week 11	Recursion, Static Variables & Mathematical Functions
Week 12	Revision
Week 13	Revision, Exam preparation
	Assignment 2 due

COMP1911 versus COMP1917

This course is intended for students who are not computer science majors. If you are a computer science major you should not be enrolled in this course unless you have a good reason, which you have discussed with your degree coordinator. Computer science majors and others who have a keen interest in computer science, and think they may take further COMP courses should enrol in COMP1917 instead! If you have previous programming experience - {\bf and enjoyed it} - choose COMP1917 A bridging course will be offered to 14s1 COMP1911 students in the midyear break. It will cover the material in COMP1917 which is not covered in COMP1911. Satisfactory performance in the exam at the finish of this bridging course will allow students to enrol in courses such as COMP1927 which have have COMP1911 as a prerequisite.

Approach to learning and teaching

This course has a heavy practical orientation. Lectures will revolve around live demonstrations of programming and use of tools. Labs & assignments form a key part.

Lectures

Lectures will be used to present the theory and practice of the techniques and tools in this course. There will be extensive use of practical demonstrations during lectures. Lecture notes will be available on the course web pages before each lecture.

Tutorials

From week 2 you will also be expected to attend a one-hour tutorial session to clarify ideas from lectures and work through exercises based on the lecture material. You should make sure that you use them effectively by examining in advance the material to be covered in each week's tute, 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. There are no marks for tutorial attendance.

Laboratory Classes

Ffollowing the tute class each week, there will be a two-hour lab class, during which you will work on a variety of small practical problems involving the tools introduced in lectures. Because this course is practical in nature, lab class are a very important component, and you should make every effort to attend the labs and complete the exercises diligently. In particular, **keep up-to-date** with the Lab work; if you fall behind it affects your ability to understand later material in the course.

To obtain a mark for a lab exercise you must both demonstrate the completed lab exercise to your tutor during a lab class and submit it using give.

If you don't complete a lab exercise during the scheduled class, you can still obtain the mark if you both submit the completed exercises before midnight Sunday and you demonstrate it to you tutor in the follow week's lab.

Lab exercises will be assessed using the following grade system:

Grade	Criteria
A	Complete, correct, and clear solution - if there are optional questions a good attempt on these must be made
B	Competent solution to core part of lab - any problems or bugs must be minor
C	Incomplete solution, or complete with significant defects or obvious bugs
D	Partial solution only, minimal achievement; or inability to explain solution
.	Not attempted

Optional extension exercises may be specified for some Labs.

Assignments

There are two assessable programming assignments. Assignments give you the chance to practice what you have learnt on relatively large problems (compared to the small exercises in the labs).

Assignments are a very important part of this course, therefore it is essential that you attempt them yourself.

- Assignment 1 (Submission, Week 8) 13%
- Assignment 2 (Submission, Week 12) 17%

These weighting and deadline may be slightly varied when the assignment designs are complete and to fit best with the lecture schedule.

Late programming assignments will be penalized. The exact penalty will be specified in the assignment specification - ften it is 2% reduction in maximum mark for every hour late.

Exam

There will be a three-hour primarily practical exam, to be held in the CSE labs during the exam period. It consists of small implementation tasks which will require you to write C programs, and a written section.

During this exam you will be able to execute, debug and test your answers. The implementation tasks will be similar to those encountered in lab exercises

It is a hurdle requirement for this course that you perform satisfactorily on the implementation tasks in the exam. Satisfactory performance is defined as solving two of the small implementation tasks.

The exact exam format (exam skeleton) will be released in week 13 but it is likely there will be 5 or 6 implementation tasks.

Assessment

Component	Value
Lab Work	10%
Assignments	30%
Exam	60%

These assessment weights might be varied by a few percent when the assignments have been chosen.

As mentioned above, your performance on the practical component of the final exam must also be satisfactory to pass the course.

The lecturer may scale overall marks, or individual components, up or down to obtain a desired mark distribution.

Course evaluation and development

Every semester, COMP1911 Student feedback is requested in a survey at the end of this course using UNSW's CATEI online survey system.

This feedback is used to improve the course materials & their delivery. Feedback from the most recent session will result in some changes to lecture delivery and assignment design,

Other matters

- Occupational Health and Safety policies
- Information for students with disabilities

Talk to the lecturer if you have any disabilities that may affect this course.

Resources for Students

The optional textbook for the course is:

- Alistair Moffat, Programming, Problem Solving, and Abstraction with C,
  Pearson Education, 2012.

For pointers to other useful reading material, including documentation for all of the tools used in the practical work, see the course Web pages.

Academic honesty and plagiarism

What is Plagiarism?

Plagiarism is the presentation of the thoughts or work of another as one's own.* Examples include:

- direct duplication of the thoughts or work of another, including by copying material, ideas or concepts from a book, article, report or other written document (whether published or unpublished), composition, artwork, design, drawing, circuitry, computer program or software, web site, Internet, other electronic resource, or another person's assignment without appropriate acknowledgement;
- paraphrasing another person's work with very minor changes keeping the meaning, form and/or progression of ideas of the original;
- piecing together sections of the work of others into a new whole;
- presenting an assessment item as independent work when it has been produced in whole or part in collusion with other people, for example, another student or a tutor; and
- claiming credit for a proportion a work contributed to a group assessment item that is greater than that actually contributed.

For the purposes of this policy, submitting an assessment item that has already been submitted for academic credit elsewhere may be considered plagiarism. Knowingly permitting your work to be copied by another student may also be considered to be plagiarism.

Note that an assessment item produced in oral, not written, form, or involving live presentation, may similarly contain plagiarised material.

The inclusion of the thoughts or work of another with attribution appropriate to the academic discipline does not amount to plagiarism.

The Learning Centre website is main repository for resources for staff and students on plagiarism and academic honesty. These resources can be located via: www.lc.unsw.edu.au/plagiarism

The Learning Centre also provides substantial educational written materials, workshops, and tutorials to aid students, for example, in:

- correct referencing practices;
- paraphrasing, summarising, essay writing, and time management;
- appropriate use of, and attribution for, a range of materials including text, images, formulae and concepts. Individual assistance is available on request from The Learning Centre. Students are also reminded that careful time management is an important part of study and one of the identified causes of plagiarism is poor time management. Students should allow sufficient time for research, drafting, and the proper referencing of sources in preparing all assessment items.

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 multiply-submitted work.

1. Submitting part or all of other students' work, with or without acknowledgement, is not acceptable.
2. Submitting solutions written by other persons is also not acceptable.
3. Building on ideas and partial solutions obtained from public sources, such as web resources, may be acceptable, provided full acknowledgement is made. However, the final mark will take into account the starting point and how much development work would have been required. Failing to acknowledge web or other resources is unacceptable.
4. Discussing approaches to solutions with other students is quite appropriate, but any discussions should remain at the design level, and must not include program text. Comparison tools will detect any common code across the student body.
5. The safest approach is to work diligently on your own, seeking help from the forum or course staff.

- Submission of work derived from another person, or jointly written with someone else will, may result in automatic failure for COMP1911 with a mark of zero.
- Allowing another student to copy from you will may result in a mark of zero for your own assignment or lab exercises. Do not provide your work to any other person, even people who are not UNSW students. You will be held responsible for the actions of anyone you provide your work to.
- Severe or second offences will result in automatic failure, exclusion from the university, and possibly other academic discipline.

Refer also to the Yellow Form material on plagiarism and the Learning Centre website