Last Updated: 11/19/2021 12:07:48

COMP1911 Computing 1A

Course Outline

Semester 2, 2012

Contents

1 Aims
2 Learning Outcomes
3 COMP1911 vs COMP1917
4 Course Information Summary
5 Reading Materials
6 Lectures
7 Tutorials
8 Laboratory Classes
9 Assignments
10 Late Submission
11 Plagiarism
12 Overall Assessment
13 Supplementary Exams
14 Getting Help
15 Course Feedback and Improvement
16 Communication with students
17 Revision History

1 Aims

The objective of this course is for students to develop proficiency in programming in the C programming 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
- the C programming language the C compiler
- programming style
- program design and organisation concepts
- program testing concepts
- program debugging concepts
- the C debugger
- program documentation
- systematic problem solving

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

2 Learning Outcomes

After completing this course, students should be able to design, construct, document and test reasonably complex C programs that accurately comply with a given specification. Further, more advanced, C programming topics will be covered in the follow-up course, COMP1921. In addition, students should have also acquired some familiarity with the UNIX/Linux operating system and tools, the C compiler, and the C debugger.

3 COMP1911 vs 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 should enroll in COMP1917 instead!

4 Course Information Summary

Course Code	COMP1911
Units of Credit	6
Hours per week	6
Prerequisite	none
Lectures	Tue 1500-1600 Wed 1000-1100

Wed 1700-1800

Location Elec Eng G24

Staff Dr. Salil Kanhere
Dr. Lavy Libman

Handbook Entry http://www.handbook.unsw.edu.au/undergraduate/courses/2012/COMP1911.html

The course website address is:

- Course Front Page: http://www.cse.unsw.edu.au/~cs1911
- Course Main Page: http://www.cse.unsw.edu.au/~cs1911/12s2

Important: You must check this site regularly for updates and announcements. If you miss an important announcement because you did not check the class website, then you will have to wear any consequences that may result. No special dispensation will be granted as a result.

5 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.
- Paul Davies, The Indispensable Guide to C, Addison Wesley, England, 1995, ISBN 0-201-62438-9.

6 Lectures

	Time	Location
Tue	1500-1600	Elec Eng G24
Wed	1000-1100	Elec Eng G24
Wed	1700-1800	Elec Eng G24

The following table is a tentative schedule of the lecture topics and assessment tasks. Students will be informed of any changes during the lectures and by announcements on the class website notice page.

Week	Lecture Topics	Assessment Tasks	Lab and Tutes
1	Course Intro, Intro to Computers/Programming		
2	Constants, Variables, printf, scanf, Expressions, if then else		Lab & Tute 1
3	Developing, testing, debugging programs		Lab & Tute 2
4	Loops, ...	Self-learning activity	Lab & Tute 3
5	Arrays, Functions		Lab & Tute 4
6	Sorting Arrays	Assignment 1	Lab & Tute 5
7	chars, strings and arrays of strings		Lab & Tute 6
	Mid Semester Break (1 Sep - 9 Sep)		
8	chars,strings,arrays of strings, debugging		Lab & Tute 7
9	Pointers, Command Line Arguments		Lab & Tute 8
10	File Operations		Lab & Tute 9
11	Recursion, Header Files, Makefiles		Lab & Tute 10
12	Revision, Exam preparation	Assignment 2	Lab & Tute 11
13	No Lectures		Lab & Tute 12

7 Tutorials

Tutorials give you a chance to clarify ideas mentioned in lectures and to practise 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 semester. There will often be a couple of extra questions, which are not covered during the tutorial. You may use these for practice or revision. Each tutorial will be worth 2 marks. 1 mark will be reserved for attempting the tutorial problems before hand and 1 mark will be reserved for active participation.

8 Laboratory Classes

See here for a location map of the Computer Science and Engineering computing labs.

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 tutor can will ask you questions about the exercises or ask you to carry out a related task.

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. Each lab exercise is worth 3 marks. There will be some opportunities to get some bonus marks. Note that, the total lab marks will be capped. Though, any surplus lab marks may replace any missing tutorial active participation marks. The lab marks are allocated as follows -

- 1 mark for preparedness
- 1 mark for correctly answering a lab prep question (question based on topics relevant to the lab exercises)
- 1 mark for quality of solutions submitted (code must be submitted using "give" at the end of the lab)

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 Friday in the week of the lab. Your tutor will then mark the submitted code by next week's lab. Only the submitted code can be assessed.

Note: 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. 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).

9 Assignments

There is one self-learning task and two assessible 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.

In the self learning task, we will assign a programming task that you will complete and submit. Following submission we will carry out a peer review of your submission so that everybody gets a chance to read and critique several other pieces of work. Your tutor will also provide feedback. This will lead to an in tutorial discussion about coding style and software development. The learning outcome is that you improve your understanding of the coding process through discussions with others and your tutor about the challenges of developing software. We would like to encourage self learning, reflection and discussion before we commence any formal assessment. This gives you a chance to learn and practise, without being penalised at this early stage of your learning.

In the first assignment, you will be given a programming problem that you will need to solve and understand how to construct the software. Instead of assessing the specific code outcome, we will assess you on your understanding of the concepts in a lab exam.

Finally, the second assignment is a major programming task. In this assignment we will expect that you are able to utilise many of the skills taught in this course. This is your chance to demonstrate that you have achieved an acceptable level of competancy in C programming.

Task	Weighting
Self Learning Task (Week 4)	0%
Assignment 1 (Lab Exam, Week 6 or 7)	15%
Assignment 2 (Submission, Week 11 or 12)	20%

The percentages represent percentages of your final mark for this course.

10 Late Submission

It is often difficult for students to catch up once they have fallen behind. So you are strongly encouraged to complete all of your programming assignments on time.

Late programming assignments will be penalised unless you have legitimate reasons to convince the LIC otherwise. Any assignment submitted after the due date will attract a reduction of 10% per day to the maximum mark. A day is defined as a 24-hour day and includes weekends and holidays. Precisely, an assignment that is submitted X hours after its due date is considered to be $\lceil X/24 \rceil$ (the smallest integer that is greater or equal to X/24) days late. No assignment will be accepted more than five days after its due date.

In the event of an illness or a family emergency that makes it impossible for you to complete an assignment on time, extensions may be granted beyond its due date. To request an extension for an assignment, contact the LIC no later than 24 hours before the assignment is due.

11 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. This software is a lot better at detecting similarities than you might expect!

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 e-mail 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.

12 Overall Assessment

The assessable components of the course are:

Component	Mark
Tutorial and Lab Mark	10%
Assignments	35%
Practical Exam	55%

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

To pass this course, students must pass the Practical Exam (i.e. achieve at least 50% of the marks), which will test your ability to do some simple programming at a lab computer.

Your final mark will be computed by adding all of the above components.

FinalMark = Tutorial and Lab Mark (10%) + Assignments (35%) + Practical Exam (55%)

The exception being that if your practical exam mark is less than 50%, then your final mark for the subject will be your exam mark (and you would fail the course).

Where your practical marks are significantly higher than your exam mark, your practical marks may be reduced and you may be required to attend an interview to explain the difference.

Any of the above may be scaled to ensure that the subject Pass/Fail boundary and the Distinction/High Distinction boundary reflect a consistent standard from session to session.

13 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.

14 Getting Help

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. Please note that as you move up the chain the number of students that person looks after increases and the likelihood of receiving an immediate response decreases.

1. check the Notices section of the class web page.
2. ask a question on the course forum
3. ask your tutor
4. attend a consultation session

For administrative issues please contact the subject administrator.

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

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

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

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

If you are new to the computing laboratory and its setup, useful documentation can be found at this link:

- http://taggi.cse.unsw.edu.au/FAQ

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 getting a copy of cseubuntu or cseXbuntu, which is available from the CSG Help Desk or can be downloaded from:

- cseXbuntu
- cseubuntu

One final piece of advice, if you have a problem, do not wait until Week 12 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.

15 Course Feedback and Improvement

The previous CATEI evaluation indicated that students were generally happy with the way the course was run but would have preferred more examples in the lectures. As a result, we are maintaining the same structure during the current session. Though, the lectures will present more hands-on programming examples.

16 Communication with students

In addition to the UNSW email address, the School of CSE provides students with a login account and email. You are hereby advised that:

- It is the policy of the School that all contact and announcements will be made via your official CSE email address only; and
- The class web page will be the other main source of announcements and information.

You should check the class web page regularly for updates and announcements. If you miss an important announcement because you did not check the class website, no special dispensation will be granted.

Students who choose to use other email accounts do so at their own risk as they may miss important announcements.

17 Revision History

9 July 2012
: Created, originating from 12s1