Course Outline

Contents

- Course Details
- Course Summary
- Prerequisites/Background Knowledge
- Course Aims
- Course Learning Outcomes
- Communication with Course Staff
- Comparison between COMP1511 and COMP1911
- Teaching Strategies and Rationale
- Student Conduct and Academic Integrity
- Assessment
- Course Schedule
- Resources for Students
- Course Evaluation and Development

Course Details

Course Codes	COMP1511
Course Title	Programming Fundamentals
Units of Credit	6
Course Website	http://cgi.cse.unsw.edu.au/~cs1511/
Handbook	COMP1511
Course Staff	**Lecturer in Charge / Course Convenor:** Marc Chee
	Lecturer: Andrew Taylor
	Course Admin: Tom Kunc
Course Contact Email	`cs1511@cse.unsw.edu.au`
Lecture Stream A (Marc Chee)	Tuesday 09:00-11:00: Central Lecture Block 7 (K-E19-104)
	Wednesday 11:00-13:00: Central Lecture Block 7 (K-E19-104)
Lecture Stream B (Andrew Taylor)	Tuesday 09:00-11:00: Physics Theatre (K-K14-19)
	Wednesday 11:00-13:00: Ainsworth G03 (K-J17-G03)
Tut-lab	3 hour slot selected when you enrol (weeks 1-5, 7-10)
Workload	In class: 7 hours/week Outside class: 4+ hours/week (a lot of optional extras), Assignments roughly 40 hours total.

Course Summary

This course introduces students to the basics of programming. Topics covered include:

- Fundamental programming concepts
- Introduction to Computer Science
- The C programming language and use of a C compiler
- Programming style
- Program design and organisation concepts
- Program testing and debugging

Prerequisites/Background Knowledge

COMP1511 has no prerequisites, and assumes no background knowledge.

Course Aims

The course aims for students to become proficient in programming using a high level language, C. By the end of the course, students should be able to construct C programs to solve problems.

Course Learning Outcomes

- understand the core syntax and semantics of the C programming language including types, I/O, arrays, functions, pointers, structs and dynamic memory allocation
- given a problem, solve it by proficiently constructing (designing, testing, debugging) a secure, reliable and correct C program
- understand and employ fundamental data structures including arrays and linked lists
- use Linux and Unix-like operating systems to develop and test software

Communication with Course Staff

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 UNSW email address. It's essential you read this email address regularly. If you forward your email, please be careful to do so correctly and test the forwarding.

Additional information will be provided in the Course Forum. You should check the course forum regularly. It is the best place to ask questions about the course.

Consultations times and Help Sessions vary through session and are listed on the course home page.

COMP1511 versus COMP1911

COMP1511 is designed for computer science majors, and for any student with a keen interest in computing, regardless of their degree program. In Term 1 COMP1511 is for computer science (and other computing) majors only.

COMP1911 is for students who are not computer science majors. COMP1911 covers a subset of the material in COMP1511 and moves at a more gentle pace. It is offered in Term 2.

If you are a computer science major, you must take COMP1511.

If you are not a computer science major, but have an interest in computer science and think you may take further COMP courses, you also should enrol in COMP1511.

And if you have previous programming experience - **and enjoyed it** - choose COMP1511

Teaching Rationale & Strategies

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

On top of this, the course is not just about the specific technical aspects of Programming, but also a preparation for studying Computer Science and the thought processes and skills necessary for a career in the field.

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.

All lectures will be recorded. UNSW's lecture recording system can only be accessed via Moodle. Moodle will not be used for any other course activities.

Lectures in Week 6 are optional.

Tutorials

From week 1 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 tutorial, by asking questions, by offering suggestions and by generally participating. The tutorial questions will be linked to on WebCMS3 in the week before each tutorial. There are no marks for tutorial attendance.

There are no tutorials in Week 6.

Laboratory Classes

Following the tutorial class each week, there will be a two-hour laboratory 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, laboratory classes are a very important component. If you do not put a great deal of effort into the lab classes you risk failing the final exam.

Each week, there will be several exercises to work on. These exercises will be released in the week preceding the lab class.

Most lab exercises will be done in pairs, and you and you partner can work through them together, learning from each other.

Tutors will facilitate you forming pairs in your week 1 lab (which is not assessed). The pairs will change twice during session.

Starting week 4, pairs will also be asked to do code reviews in the tutorials, to explain how they tackled a particular problem and describe interesting features of their solution. There are no laboratory classes in Week 6.

Overview

Lab exercises will be automarked (marked automatically by a computer), so that tutors can spend lab time assisting students rather than marking labs.

Submission

For each of the lab exercises, **both** members of each lab pair need to submit the exercise **separately** using `give` .

You cannot obtain marks by e-mailing lab work to tutors or lecturers.

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 **before 5pm Sunday** (Sunday 17:00) in the same week as the lab.

Challenge Exercises

Challenge exercises may be specified for some labs.

Some challenge lab exercises typically will specify that they are individual exercises (not to be done with your partner).

Challenge exercises may be silly, confusing or unreasonably difficult.

Do not worry if you can not complete challenge exercises, you can receive full marks for labs without attempting any challenges. Challenges give 'bonus marks' which you can use to make up for lost lab marks.

Lab Marking

Lab exercises will be automarked, using test cases that you haven't seen: different to the test cases `autotest` runs for you. (Hint: do your own testing as well as running `autotest`)

There will be partial marks for attempts which fail some of these automated tests.

Automarking

Automarking will be run several days after the submission deadline for each lab. When it is complete you can see marks online: view it here or by running this command on a CSE machine:

```
1511 classrun -collect exercise_name
```

Lab Marks

When all components of a lab are automarked you should be able to view the the marks via give's web interface or by running this command on a CSE machine:

```
1511 classrun -sturec
```

There will be more lab marks available than necessary to obtain full marks for the 10% lab component. In other words: total lab marks will be capped.

The lab class for each week is worth in total 1.4 marks.

There are no marks for the week 1 lab, it is there to help you get started.

Usually each lab exercise will be worth the same - for example if there are 4 lab exercises each will be worth 0.35 marks.

Challenge exercises will award marks that go beyond the maximum mark. These marks will only be added to your total marks for all labs if they are less 10.

The best 7 of your 8 lab marks for weeks 2-10, will be summed to give you a mark out of 10.

If their sum exceeds 10 - your total mark will be capped at 10.

Hence:

- You can miss 1 lab without affecting your mark.
- You can obtain full marks for the lab component without doing any challenge exercises.

Help Sessions

There will be consultation sessions starting in week 2 where tutors will be available for one on one help with specific problems and assignment clarification. These sessions are optional and will run at different times during the week, with more sessions available around assignment deadlines and in later weeks of the term. Check the course timetable for what Help Sessions have been scheduled.

Live Streaming

Live Streaming sessions that you can join from your own computer will be run at specific times in the term when help is needed. These will involve some live problem solving and coding that will answer frequently asked questions as well as show some more examples of specific coding based on student requests. Live participation will allow students to ask questions during the sessions. These sessions will be recorded and available for viewing after they're streamed. Check the course website for a stream schedule.

Weekly Coding Tests

There will be 7 weekly coding tests from weeks 3-5 and 7-10 designed to give you timely & realistic feedback of your understanding of the course material.

These are conducted in two stages:

The first stage is in your own time under self-enforced exam-like conditions. This gives you both accurate feedback on your progress as well as some practice for coding under time constraints.

Each test will specify the conditions but typically these will include:

1. No assistance from any person.
2. A time limit (1 hour).
3. No access to materials (written or online) except specified language documentation or man pages.

After you've completed that stage, you will have a reasonable idea of how you're going in the course at this time and what topics you might want to study further.

The second stage is treating the test like an exercise and continuing it with whatever resources you'd like to use (like a lab exercise). The only difference is: We won't be discussing the weekly tests on the forums until after everyone's had a chance to complete them. We don't want to spoil other people's feedback by giving them the answers too early.

Each coding test will be automatically marked. There will be partial marks for attempts which do not pass automatic tests. Here is an indicative guide:

Passes all automatic tests.	1/1
Fails several automatic tests.	0.75/1
Any attempt at the test	0.5/1

Your mark for the coding test component will be the sum of your best 6 of 7 test marks.

Any deliberate violation of the test conditions will result in a mark of zero for the entire programming test component.

The weekly programming test must be completed by Thursday 5pm the week after it is released.

Assignments

There are two assessable programming assignments. Assignments give you the chance to practice what you have learned 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. Collaboration with other students is limited to discussion of fundamentals, not any discussion of assignment specifics.

- Assignment 1 (Submission, Week 6) 15%
- Assignment 2 (Submission, Week 10) 20%

The assignment weighting and deadlines may change a little when the assignment designs are complete.

Late assignments submissions will be penalized. The exact penalty will be specified in the assignment specification - typically it is 1% reduction in maximum mark for every hour late.

Final Exam

There will be a three-hour primarily practical exam, to be held in the CSE labs during the exam period. This will be centrally timetabled and appear in your UNSW exam timetable.

It will contain implementation tasks which will require you to write C programs. It will also contain sections which require you to read code or answer questions to show your knowledge of programming.

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

Exam Requirements

COMP1511 has two requirements on the final exam.

Requirement #1: on the final exam you must solve a task by writing a program that uses an **array**. The final exam will contain multiple questions (clearly marked) which if answered successfully will meet this requirement. Answering any one of these questions will meet this requirement.

Requirement #2: on the final exam you must solve a task by writing a program that uses a **linked list**. The final exam will contain multiple questions (clearly marked) which if answered successfully meet this requirement. Answering any one of these questions will meet this requirement.

You can not pass COMP1511 unless you achieve both the above requirements.

Assessment

Component	Weight
Lab Work	10%
Weekly Tests	5%
Assignments	35%
Final Exam (exam period)	50%

Supplementary Assessment

Students will be offered a supplementary exam if they miss the original exam due to (documented) illness or misadventure. Applications for Special Consideration are handled by UNSW Student Support and Services, not by subject staff.

The supplementary exam is scheduled for the week of the 25-29th May 2020. It is your responsibility to be in Sydney and available for the supplementary exam. No alternative will be offered.

Student Conduct

The Student Code of Conduct (Information, Policy) sets out what the University expects from students as members of the UNSW community.

As well as the learning, teaching and research environment, the University aims to provide an environment that enables students to achieve their full potential and to provide an experience consistent with the University's values and guiding principles. A condition of enrolment is that students inform themselves of the University's rules and policies affecting them, and conduct themselves accordingly.

In particular, students have the responsibility to observe standards of equity and respect in dealing with every member of the University community. This applies to all activities on UNSW premises and all external activities related to study and research. This includes behaviour in person as well as behaviour on social media, for example Facebook groups set up for the purpose of discussing UNSW courses or course work.

Behaviour that is considered in breach of the Student Code Policy as discriminatory, sexually inappropriate, bullying, harassing, invading another's privacy or causing any person to fear for their personal safety is serious misconduct and can lead to severe penalties, including suspension or exclusion from UNSW.

If you have any concerns, you may raise them with your lecturer, or approach the School Ethics Officer, Grievance Officer, or one of the student representatives.

All work submitted for assessment must be your own work.

Lab exercises must be completed by you and your partner.

Assignments must be completed *individually*.

Submission of other people's work as your own (plagiarism) has a major impact on learning so we use plagiarism detection software to search for multiply-submitted work.

Please note:

- Submitting part or all of other students' work, with or without acknowledgement, is not acceptable.
- Submitting work written even partly by another person is also not acceptable.
- Building on ideas and code snippets obtained from public sources, e.g. Stack Overflow, is acceptable (unless the assignment/lab forbids this) provided full acknowledgement of their resources is made.
- Discussing approaches to exercises and assignments 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 and previous iterations of the course.
- 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 result in loss of marks and your name being entered on UNSW's plagiarism register. This loss of marks often results in the student failing the course.
- Providing your work to another person will also result in the loss of marks and your name being entered on UNSW's plagiarism register. 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 may result in automatic failure or exclusion from UNSW.

Make sure you read:

- the Learning Centre's explanation of plagiarism.
- UNSW's policy regarding academic honesty and plagiarism

Other matters

- Occupational Health and Safety policies
- Information for students with disabilities
  Contact the lecturer ASAP if you have any disabilities that may affect this course.

Course Schedule

Week 1	Course Introduction, Compiling and Running C Programs, Variables, Simple input and output, If statements
Week 2	Continuing If statements, Boolean Expressions, Loops, Problem Solving
Week 3	Nested Loops, Memory and Arrays, Functions, Code Reviews
Week 4	Pointers and Memory, Code Style
Week 5	Characters, Strings, Structs, Debugging and Testing
Week 6 (Flexibility Week)	Guest Lectures, Halfway course recap, Professionalism **Assignment 1** due
Week 7	Multi-file projects, Memory Allocation, Linked Lists
Week 8	Linked Lists
Week 9	Abstract data types, Queues and Stacks,
Week 10	Revision, Exam preparation **Assignment 2** due
Exam Period	**Final Exam**

Resources for Students

No textbook is required for COMP1511.

The optional textbook for the course is: *Programming, Problem Solving, and Abstraction with C* by Alistair Moffat, ISBN 978 1 74103 080 3, which can be purchased from the UNSW Bookshop.

Course Evaluation and Development

Every trimester, COMP1511 student feedback is requested in a survey at the end of this course using UNSW's myExperience online survey system.

This feedback is used to improve the course materials and their delivery.

In 2019 feedback was very favourable probably as a result of changes based on the previous session's feedback. Feedback from surveys for COMP1511's previous offerings has resulted in changes to COMP1511 delivery, including increasing Help Sessions and Live Streaming as well as updating Weekly Tests and Assignments.

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.

CSE may also run its own survey, midway through the term, to ellicit feedback while courses are still running. This course improves only because we see the difficulties that students have and try to adjust things so that you get to learn what you need. If anything's not working for you, please let us know and we'll do whatever we can to help and hopefully help students in later cohorts as well.

COMP1511 20T1: Programming Fundamentals is brought to you by
the School of Computer Science and Engineering
at the University of New South Wales, Sydney.
For all enquiries, please email the class account at cs1511@cse.unsw.edu.au
CRICOS Provider 00098G