Course Outline

Contents

- Course Details
- Course Summary
- Assumed Knowledge
- Student Learning Outcomes
- Teaching Strategies and Rationale
- Student Conduct
- Academic Honesty and Plagiarism
- Assessment
- Course Schedule
- Resources for Students
- Course Evaluation and Development

Course Details

Course Code	COMP1511
Course Title	Introduction to Programming
Convenor	John Shepherd
Lecturer	Andrew Bennett
Admin	Mei Cheng Whale
Classes	**Lectures** : Mon 12-2 Science Theatre, Wed 4-6 Science Theatre Timetable for all classes (/COMP1511/17s2/timetable)
Consultations	Tuesday 2:30pm - 3:30pm, K17 Ground Floor Consultation Rooms
Units of Credit	6
Course Website	http://cse.unsw.edu.au/~cs1511/17s2/ (http://cse.unsw.edu.au/~cs1511/17s2/)
Handbook Entry	http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP1511.html (http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP1511.html)

Course Summary

Welcome to COMP1511! This course is an introduction to solving problems, and the craft of designing and implementing elegant software solutions. Along the way, we'll introduce fundamental concepts of programming and of software development, the C programming language, and programming in teams. We look forward to joining you on your first computing adventures!

Assumed Knowledge

COMP1511 has no prerequisites, and assumes no background knowledge or prior programming experience.

Student Learning Outcomes

After completing this course, students will be able to:

1. design complete software solutions for simple problems
2. work in a team to develop software
3. test and debug programs
4. create and use simple data structures
5. have a solid understanding of values, storage and addressing
6. understand and appropriately use abstraction
7. write programs using good programming style
8. distinguish between well-written programs and poorly written programs
9. design software solutions for larger problems using abstraction and interfaces
10. reflect on their programming and develop the ability to continually improve their skills

This course contributes to the development of the following graduate capabilities:

Graduate Capability	Acquired in
Scholars capable of independent and collaborative enquiry, rigorous in their analysis, critique and reflection, and able to innovate by applying their knowledge and skills to the solution of novel as well as routine problems	1-10
Entrepreneurial leaders capable of initiating and embracing innovation and change, as well as engaging and enabling others to contribute to change	
Professionals capable of ethical, self- directed practice and independent lifelong learning	
Global citizens who are culturally adept and capable of respecting diversity and acting in a socially just and responsible way	

Teaching Strategies and Rationale

As with playing a musical instrument, computer programming cannot be learnt by simply watching and listening. Hands-on experience emphasised through lab work and assignments is essential in mastering the skills taught in this course.

Computing is best learned by practice, and thus labs and programming assignments are a critical component of the course. These allow students to practice design and implementation skills, and to develop teamwork skills.

Tutorials will provide a forum for students to develop design skills and to practice presentations.

Fortnightly milestone write-ups will assist in developing students ability to reflect on their own work.

Lectures

Lectures will be split between discussion of concepts, discussion of practical work (and practical demonstrations), revision work, and extension lectures.

Tutorials

Tutorials are an opportunity to clarify ideas from lectures and work through exercises based on the lecture material, and you are expected to attend these. You should make sure that you use the tutorials 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 course website before each tutorial. There are no marks for tutorial attendance.

Tutorials will start in week 1.

Lab Classes

Lab classes aim to give you practice in problem-solving and program development.

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 concepts and strategies introduced in lectures. Because this course is has a significant practical component, laboratory classes are important. If you do not put a good amount of effort into the lab classes you risk failing the final exam.

Each week, there will be one or more exercises to work on. These exercises will be released in the week preceding the lab class. Labs will be done in pairs, and you and you partner should discuss the exercises before going to the lab, to maximise the usefulness of the class. Some of the exercises must be done in pairs, while others may be done individually: this will be explicitly noted with each exercise.

The exercises will need to be submitted, and will be assessed both as part of the fortnightly milestone write-ups (see below), as well as with a series of automated tests, which will assess correctness as well as programming style. During the lab, your tutor will provide feedback on your approach to the problem and on the style of your solution.

Tutors will facilitate you forming pairs in your week 1 lab. The pairs will change several times during session.

Starting in week 2, 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.

Milestone Write-ups

There will be regular "milestones" throughout the course, where you will submit a short write-up of your progress since the last milestone. More information is available on the milestones overview page (https://webcms3.cse.unsw.edu.au/COMP1511/17s2/resources/11958) .

Assignments

There are three 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). **The first assignment is an individual assignment; the others will be completed in groups.** We expect all members of a group to contribute to the assignments; part of your assignment mark will be tied to this. As noted above, assignments are the primary vehicle for learning the material in this course. If you don't do them, or simply copy and submit someone else's work, you have wasted a valuable learning opportunity.

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

Student Conduct

The **Student Code of Conduct** (Information (https://student.unsw.edu.au/conduct) , Policy (https://www.gs.unsw.edu.au/policy/documents/studentcodepolicy.pdf)) 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 (mailto:ethics-officer@cse.unsw.edu.au) , Grievance Officer (mailto:grievance-officer@cse.unsw.edu.au) , or one of the student representatives.

Academic Honesty and Plagiarism

Plagiarism is defined as (https://student.unsw.edu.au/plagiarism) using the words or ideas of others and presenting them as your own. UNSW and CSE treat plagiarism as academic misconduct, which means that it carries penalties as severe as being excluded from further study at UNSW. There are several on-line sources to help you understand what plagiarism is and how it is dealt with at UNSW:

- Plagiarism and Academic Integrity (https://student.unsw.edu.au/plagiarism)
- UNSW Plagiarism Procedure (https://www.gs.unsw.edu.au/policy/documents/plagiarismprocedure.pdf)

Make sure that you read and understand these. Ignorance is not accepted as an excuse for plagiarism. In particular, you are also responsible that your assignment files are not accessible by anyone but you by setting the correct permissions in your CSE directory and code repository, if using. Note also that plagiarism includes paying or asking another person to do a piece of work for you and then submitting it as your own work.

UNSW has an ongoing commitment to fostering a culture of learning informed by academic integrity. All UNSW staff and students have a responsibility to adhere to this principle of academic integrity. Plagiarism undermines academic integrity and is not tolerated at UNSW. Plagiarism at UNSW is defined as using the words or ideas of others and passing them off as your own.

If you haven't done so yet, please take the time to read the full text of

- UNSW's policy regarding academic honesty and plagiarism (https://student.unsw.edu.au/plagiarism)

The pages below describe the policies and procedures in more detail:

- Student Code Policy (https://www.gs.unsw.edu.au/policy/documents/studentcodepolicy.pdf)
- Student Misconduct Procedure (https://www.gs.unsw.edu.au/policy/documents/studentmisconductprocedures.pdf)
- Plagiarism Policy Statement (https://www.gs.unsw.edu.au/policy/documents/plagiarismpolicy.pdf)
- Plagiarism Procedure (https://www.gs.unsw.edu.au/policy/documents/plagiarismprocedure.pdf)

You should also read the following page which describes your rights and responsibilities in the CSE context:

- Essential Advice for CSE Students (https://www.engineering.unsw.edu.au/computer-science-engineering/about-us/organisational-structure/student-services/policies/essential-advice-for-cse-students)

Assessment

Overall weighting

Item	What	Total Marks
Milestone write-ups	Fortnightly write-ups	10%
Assignments	Three assignments across the semester	30%
Labs	Weekly lab exercises	5%
Exams	Practical and theoretical questions	55%

Individual assessments

Item	Topics	Due	Marks	Contributes to
Milestone write-ups	All topics	Fortnightly -- see schedule below	10%	1-10
Assignment 0		(end of) Week 5	5%	1, 3, 7, 10

Assignment 1		(end of) Week 7	10%	1, 2, 7, 10
Assignment 2		Week 12	15%	1, 2, 3, 6, 7, 9, 10
Labs	All topics	All Weeks	5%	1-10
Prac exam #1			10%	1, 3, 4, 5, 7,
Prac exam #2			10%	1, 3, 4, 5, 6, 7, 8,
Final Exam		Exam period	35%	1, 3-9

Hurdles

COMP1511 has two exams hurdle requirements:

Under exam conditions show that you can solve tasks by writing programs that use **arrays** . You will have three opportunities to do this: the week 7 practice practical exam, the week 10 practical exam, and the final exam. You only need pass the hurdle in one of these three exams.

Under exam conditions show that you can solve tasks by writing programs that use **linked lists** . You will have two opportunities to do this: the week 13 exam or the final exam. You can pass the hurdle in one of these two exams.

You can not pass COMP1511 unless you achieve both the above hurdles. You will be offered an additional chance to pass the hurdles in the supplementary exam, if you achieve a mark of 50+ but do not pass the hurdles.

Supplementary Assessment

Students will be offered a supplementary exam if they miss the original exam due to (documented) illness or misadventure.

Students who will be automatically offered supplementary assessment if they achieve a final mark of 50+ but fail to meet the hurdle requirement, if they have attended 9+ labs and have made reasonable attempts on both assignments (achieving > 50%)

Students with final marks in the range 40-49 (whether they have met the hurdle requirement or not) will also be offered supplementary assessment if they have attended 9+ labs and have made reasonable attempts on both assignments (achieving > 50%)

Course Schedule

This schedule is **indicative** and may be subject to change.
You should check each task's due dates, and not rely solely on dates listed below.

Week	Lectures	Exams	Assignments	Notes
1	Course intro, problem solving, abstraction, linux and shell basics, first C program, compiling and running C programs	-	-	-
2	If statements, logic, logical operators, bits binary and hexadecimal, design with functions	-	-	-
3	Functions, memory, types	-	-	Milestone 1 write-up
4	Loops, arrays	-	-	-
5	Pointers, strings	-	Assignment 0 due	-
6	Structs, run-time stack, stack frames, buffer overflows	-	-	Milestone 2 write-up
7	More memory, malloc, heap	Practice Prac exam #1	-	-
8	Typedef, concrete vs abstract, ADTs	-	Assignment 1 due	*(no milestone: final assn1 reflection instead)*
9	ADTs	-	-	-
	midsem break	-	-	Milestone 3 write-up
10	*quiet week - no classes*	Prac exam #1	-	-
11	Linked lists	-	-	Milestone 4 write-up

12	Stacks and queues	-	-	-
13	Professionalism, ethics	Prac exam #2	Assignment 2 due	Final Milestone write-up
Study break		-	-	-
Exam period		Final exam	-	-

Resources for Students

No course textbook is required.

The following text may be valuable as an additional resourcs:

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

Other resources:

- http://www.cprogramming.com/tutorial/c-tutorial.html (http://www.cprogramming.com/tutorial/c-tutorial.html)
- http://www.tutorialspoint.com/cprogramming (http://www.tutorialspoint.com/cprogramming)

Course Evaluation and Development

This course is evaluated each session using the myExperience system.

In the previous offering of this courses, students noted various concerns regarding the course's structure and coverage of content.

Based on their comments, we have restructured the course content and coverage, as well as assessment structure.

Resource created 4 years ago (Tuesday 18 July 2017, 11:30:24 PM), last modified 4 years ago (Monday 25 September 2017, 06:08:58 PM).

Comments (/COMP1511/17s2/forums/search?forum_choice=resource/11027) (/COMP1511/17s2/forums/resource/11027)

There are no comments yet.