Resources / Course Outline

Course Outline

Contents

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

Course Details

Course Code	COMP1511
Course Title	Programming Fundamentals
Lecturer/Convenor	Andrew Taylor (/users/z9300162)
Admin	Tom Kunc (/users/z5205060)
Lectures	**Tuesday** 9am-11am online **Thursday** 2pm-4pm online Details for online lectures are below
Tutorials and Labs	Timetable for all classes (https://webcms3.cse.unsw.edu.au/COMP1511/21T2/timetable)
Help Sessions	Face to Face Help Sessions will be timetabled soon . . .
Course Contact Email	cs1511@cse.unsw.edu.au
Units of Credit	6
Course Website	http://cse.unsw.edu.au/~cs1511/21T2/ (http://cse.unsw.edu.au/~cs1511/21T2/)
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

This course is an introductory course into the basics of Computer Programming and Computer Science. It is intended as an introduction to studying further in Computer Science or related fields. Topics 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

Assumed Knowledge

Before commencing this course, students should:

- Have basic computer literacy (not necessarily have programmed before)

Otherwise COMP1511 assumes no background knowledge.

Student Learning Outcomes

This course aims for students to become proficient in a high level programming language, C. It also focuses on mental preparedness for programming long term, including problem solving, debugging and testing.

After completing this course, students will:

- Have basic proficiency with the C programming language
- Have the ability to analyse a problem and solve it using programming
- Have learnt some techniques for debugging and testing code and programs
- Understand how to use basic data structures like arrays and linked lists
- be able to use the basics of a Linux-like, command line driven operating system

Teaching Strategies

This course has a heavy practical orientation. Lectures will revolve around live demonstrations of programming and use of tools. Labs and assignments are also highly practical.

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.

The platform on which lectures will be delivered is to be finalized later this week.

All lectures will be recorded. For anyone who cannot access the live stream or who cannot access them live, lecture recordings will be made available.

Lectures in Week 6 are optional.

Live Streaming and Videos
In addition to scheduled lectures, there will be some more informal streams and pre-recorded videos to give students a chance to ask questions directly as well as cover content that is not "official" course content, but still might be very useful.

Tutorial/Lab Sessions

From week 1 you will also be expected to attend a three hour tutorial/laboratory 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 the class webpage in the week before each tutorial. There are no marks for tutorial attendance.

Tutorial/Labs will be run in two different formats:

- For classes with codes ending in an "A" (for example: T09A or F12A): classes will be face-to-face at UNSW's Kensington Campus. *These classes will be held under strict COVID-safe rules. They are only open to students officially enrolled in them via myUNSW. They may be cancelled at any time. Students who feel unwell **must not** attend these classes.*
- For classes with codes ending in "B" or "C" (for example: T09B or F12C): classes will be held online via a system called Blackboard Collaborate. Links for Tutorial/Lab sessions will be available on Moodle. Students in face-to-face classes may attend online classes if they are unable to attend in person, or if they feel unwell.

The format of online and face-to-face tutorial/lab sessions will be identical, only the method of delivery will be different.

Following the theoretical section of the Tutorial/Lab session, there will be time to work on practical exercises as well as have some time to have one on one conversations with your tutors to get specific help.

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.

Most lab exercises will be done in pairs or small groups, and you and other students can work through them together, learning from each other. Tutors will facilitate you forming pairs/groups during Laboratory Sessions.

There are also some Challenge Exercises for anyone looking to push beyond the course content or see some interesting continuation of the content in the course. They are not necessary to complete.

Lab Submission and Marking
Each lab exercise will be submitted using the "give" system. All students will need to submit solutions, even if they've worked in a group and have the same solution as someone else.

If you cannot complete any exercises by the end of the lab you may complete them in your own time and submit them using the "give" command before 8pm Monday (Monday 20:00 AEST) in the week after the lab.

Labs will be marked automatically around a week after the due date.

When marking is complete you can see marks online here (http://cgi.cse.unsw.edu.au/~cs1511/21T2/student/) or by running this command on a CSE machine:

```
1511 classrun -collect exercise_name
```

There are no marks for Lab 1, it's there to help you get started. Your total mark will be made up of the best 7 out of 8 of the labs from weeks 2-10 (there is no lab in week 6).

Challenge Exercises in labs are not necessary to complete and it is possible to obtain full marks without attempting any Challenges. Any marks from Challenge exercises will only be added to your marks if they are below the maximum total for labs.

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.

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:

- No assistance from any person, or asking questions online.
- A time limit (1 hour).

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 a lab exercise and continuing it with whatever resources you'd like to use. 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. Partial marks will be awarded based on the percentage of marking tests passed. These tests will be similar to the automated tests provided to you, but may differ slightly.

Marks for the coding test component will be the sum of the 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 8pm the week after it is released. **[A previous version of this course outline said 5pm; we have updated it to 8pm].**

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) 25%

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 2% reduction in maximum possible mark for every hour late.

Final Exam

In 2021 Term 2, the Exam will be a 6 hour long take-home exam. The actual working time of the exam is only expected to be around 3 hours, but due to time zone differences for current students, all students will be given a longer time window to complete the exam.

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 and Weekly Tests.

Special 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.

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.

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

Item	Topics	Due	Marks Released	Marks
Lab Exercises	All topics	Weeks 2-10	1 week after they are due.	10%
Weekly Tests	All topics	Weeks 3-10	1 week after they are due.	10%
Assignment 1	Looping and Arrays	Week 6	Between weeks 8 and 9.	15%
Assignment 2	Linked Lists	Week 10	**After** the final exam is sat.	25%
Final Exam	All topics	Exam period	Before the T2 holidays.	40%

Course Schedule

Week	Lectures	Tutes	Labs	Live Streams	Assignments	Weekly Test
1	Course intro, Our First C Program (variables and if statements)	Welcome, What is programming?	Lab familiarization, Setting up working from home (VLAB), Basic Input/Output	-	-	-
2	Problem Solving and Looping	Variables and If Statements	Variables and If Statements	-	-	-
3	Code Style, Code Reviews, Functions and Arrays	Looping	Looping	-	-	Basic Input/Output, if statements
4	Functions, Arrays, Memory and Pointers	Code Reviews, Functions and Arrays	Code Reviews, Functions and Arrays	Assignment 1 overview	Assignment 1 released	Looping
5	Debugging, Characters, Strings and Structs	Functions, Arrays and Pointers	Functions, Arrays and Pointers	-	-	Functions and Arrays

6	Flexibility Week. Guest Lectures, Halfway Course recap. Professionalism	No Tutorials	No Labs	-	Assignment 1 due	No Test
7	Memory Allocation, Multi-file projects and Linked Lists	Characters, Strings and Structs	Characters, Strings and Structs	-	-	Functions, Arrays and Pointers
8	Linked Lists	Memory Allocation and Linked Lists	Memory Allocation and Linked Lists	Assignment 2 overview	Assignment 2 released	Characters, Strings and structs
9	Abstract Data Types, Recursion	Linked Lists	Linked Lists	-	-	Memory and Linked Lists
10	Exam prep and Course recap	Abstract Data Types	Exam practice (past exam questions)	-	Assignment 2 due	Past exam questions
11	Revision and study for the exam			Revision stream	-	-

Resources for Students

There is no requirement for a text book for COMP1511. Generally, students do not purchase this textbook. It covers material in a different way to the COMP1511 course materials, and goes into differing levels of content than this course. It may, however, be useful as a reference, or to explore some content in more detail.

The optional textbook for the course is: *Programming, Problem Solving, and Abstraction with C* by Alistair Moffat (http://www.bookshop.unsw.edu.au/details.cgi?ITEMNO=9781486010974) , ISBN 978 1 74103 080 3, which can be purchased from the UNSW Bookshop.

Course Evaluation and Development

Every term, 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.

The feedback from 21T1 was very good for COMP1511 so we'll teach 21T2 very much in the same way. The feedback about reflections suggested their implementation didn't work well. In 21T2 we've removed them as an explicit assessment item and we'll to integrate reflection into labs.

2020 provided a very unique opportunity to test online teaching for COMP1511. In general, we found that Live Streaming was a much more favourable format than live lectures, so we'll definitely be continuing that through 2021 , even potentially focusing more on that style of interactive teaching than the old fashioned lectures.

We still have some work to do on more interactive teaching in labs, which have seen the most change in the switch to online teaching. Luckily, we've received a lot of useful feedback that we're hoping to put into effect. This term there will possibly be a combination of in-person and online tutorials, so we will hopefully learn

something useful from that combination.

Students are also encouraged to provide informal feedback during the session, and to let the course staff know of any problems, as soon as they arise.

CSE may also run its own survey, midway through the term, to elicit 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.

Resource created 6 months ago (Wednesday 19 May 2021, 10:09:55 AM), last modified 5 months ago (Tuesday 15 June 2021, 09:14:30 AM).

Comments

(/COMP1511/21T2/forums/search?forum_choice=resource/60895)

(/COMP1511/21T2/forums/resource/60895)

There are no comments yet.