UNSW Global

Course Outline

CP1511

Introduction to Programming

Diploma Program

UNSW Global Education

Term 3 2020

1. Staff

Position	Name	Email
Course Convenor & Lecturer	Dr Aarthi Natarajan	a.natarajan@unswglobal.unsw.edu.au
Course Convenor & Lecturer	Dean Wunder	d.wunder@unswglobal.unsw.edu.au

2. Course information

Units of credit (UOC):	6
Pre-requisite(s):	none
Total course contact hours:	96

2.1 Course summary

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

- fundamental programming concepts
- the C programming language and use of a C compiler
- programming style
- program design and organisation concepts
- program testing and debugging

The course does not assume any previous programming experience.

2.2 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.

2.3 Course learning outcomes (CLO)

At the successful completion of this course you (the student) should be able to:

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

2.4 Relationship between course and program learning outcomes and assessments

Course Learning Outcome (CLO)	**Program Learning Outcome (PLO)**	**Related Tasks & Assessment**
CLO 1	Conceptual understanding of computer underpinnings (EA1.2)	Exam, Practical Exams, Labs, Assignments
CLO 2	Understanding of specialist bodies of engineering knowledge (EA1.3)	Exam, Practical Exams, Labs, Assignments
CLO 3	Understanding of underpinnings (EA1.1)	Exam, Practical Exams, Labs, Assignments
CLO 4	Understanding of specialist bodies of engineering knowledge (EA1.3)	Labs, Practical Exams, Assignments

3. Strategies and approaches to learning

3.1 Learning and teaching activities

This course involves a number of teaching activities:

Lectures – 4 hours per week

Lectures present theory and concepts, by way of case studies and practical examples. Lecture notes will be provided in advance of each class. There will be 4 hours of timetabled live streamed lectures each week.

Tutorials – 1 hour per week

Tutorials allow students to collaboratively work through example problems to illustrate lecture idea, and have concepts from lectures clarified by the tutor.

Lab Classes – 3 hours per week

- Attendance will be taken for all Lab sessions.

Lab A Sessions:

- These lab classes involve small exercises where students build systems that illustrate the ideas covered in lectures. In **some** of the 2 hour Lab A classes, students may be required to work in pairs to write software. If students are required to form pairs, this will be indicated in the lab specification for that week.
- To obtain a mark for a Week X lab exercise you should **submit** the completed lab by Sunday 8:00 pm, in the same week X using **GIVE**. Late submissions will incur a penalty.
- Students will seek help from their lab demonstrators during lab A/B sessions.
- “Challenge exercises” will be provided for students who find the regular weekly exercises too easy and will allow students to earn bonus lab marks. Each lab is worth a possible 1.2 marks.
- The best 10 out of 11 lab marks are added up and capped at 10. During Lab A in week 12 a practice exam worth 2 marks will be held.
- Through the semester, we will also have regular **check-points**. These will be held in Lab A sessions, during which students will be asked questions regarding the solutions submitted during the past weeks. Dates of these check-point sessions will be announced through the semester.

Lab exercises will be assessed using the following grade system:

Grade	Criteria
1.3/1.0	Outstanding effort; Correct solution to every standard + every challenge lab exercise.
1.0/1.0	Correct solution to every standard lab exercise (worth full marks)
0.5/1.0	Correct solution to half of the standard lab exercises completed
0/1.0	No attempt or incorrect solution that fail marking tests for all exercises.

- Partial marks will be awarded for submissions which pass some but not all marking tests.
- There will be more lab marks available than necessary to obtain full marks for the 10% lab component. In other words, the total lab mark will be capped at 10.
- **You cannot obtain marks by e-mailing lab work to tutors.**

Lab B Sessions

In the 1 hour Lab B **classes, students work individually on programming. In odd weeks (3, 5, 7, 9, 11) this hour will hold assessable practical exams**

These practical exams are assessed and help prepare students for the final exam. These are held in weeks 5, 7, 9, 11 and 12. They are each worth 2 marks.

2020 Covid-19 Edition: All lectures, tutorials and labs are now being delivered entirely Blackboard Collaborate, with links to the session uploaded on Moodle. All lectures will be recorded through Collaborate.

Assignments – 2 during the course

Assignments are take-home problems that are larger in scope than Lab exercises and require students to use creativity to solve a challenging realistic problem. Each assignment requires students to understand the problem, design a solution, and implement and test their solution.

Online Forum

An online forum allows students to ask and answer questions on the tutorial, lab and assignment exercises, and on lecture material.

Final Exam

The format of the final exam is yet to be finalised. The format and mode of the exam will be advised closer to the end of the term. In the final exam, students will need to solve both theory and practical problems.

3.2 Help Sessions

There will be consultation sessions starting in week 1 where tutors will be available for one on one help with specific problems and assignment clarification. These sessions are optional and will run at a scheduled time during the week, with more sessions available around assignment deadlines and during exam study period. Check the course timetable for what Help Sessions have been scheduled.

2020 Covid-19 Edition: Help Sessions will be held online through Blackboard Collaborate. Students will still be able to receive one on one help in these sessions, but it won't be face to face like usual.

3.3 Expectations of students

Students are expected to:

- attend all lectures, and ask questions, but otherwise not disturb other students
- attend all tutorials and actively participate in the discussions
- attend all lab classes and work diligently on the exercises
- do all of the assignment work themselves, asking only the forum or tutors for help

On the course forum, students should:

- use relevant/meaningful message titles on all posts
- ask questions clearly and provide sufficient background information that the question can be reasonably answered
- not post significant pieces of code, especially code for assignments

4. Course schedule and structure

This course consists of 8 hours of class contact hours per week. You are expected to take an additional 5 hours outside classes to complete assessments, readings, and exam preparation.

Week	**Lectures**	**Tutorial and Labs**	**Assessment**	Related CLO
Week 1	Introduction to course/Linux/C; data types; variables, simple I/O, expressions, It Statements	Create/run first C programs on Linux		1,4
Week 2	Loops	arithmetic and simple control		1,2,4
Week 3	Memory and Functions	complex expressions, iteration		1,2,4
Week 4	Arrays	functions, exploring memory	Assignment 1 Released	1,2,4
Week 5	Chars, strings	array manipulation	Practical Exam 1	1,2,4
Week 6	Pointers and extra C (for loops, pre/post increment, multi-file compilation)	working with characters and strings		1,2,4

Week 7	Dynamic memory allocation, C implementation memory models, struct	command line arguments	Practical Exam 2 **Assignment 1 Due**	1,2,4
Week 8	Linked Lists	malloc, pointers, struct	**Assignment 2 released**	1,2,3,4
Week 9	Abstract Data Types, including stacks and queues	programming with linked-lists	Practical Exam 3	1, 2, 3, 4
Week 10	Recursion including recursion with linked lists	programming and testing ADTs		1, 2, 3, 4
Week 11	Introduction to searching, sorting and complexity (Not examinable)	programming with linked lists using recursion	Practical Exam 4	1, 2, 3, 4
Week 12	Exam Information and Revision	sorting and a practice exam	Practical Exam 5 **Assignment 2 due**	1, 2, 3, 4

5. Assessment

5.1 Assessment tasks

Assessment task	Length	Weight	Due	CLOs
Assessment 1: Programming assignment (arrays)	3 weeks	15%	Friday week 7	1, 2, 4
Assessment 2: Programming assignment (linked data structures)	3 weeks	15%	Friday week 12	1, 2, 3, 4
Assessment 3: Lab exercises (the best 10 out of 11 will be taken)	Throughout semester	10%	Weekly, on Sunday	1, 2, 3, 4
Assessment 4: Practical Exams	Throughout semester	10%	Weeks 5,7,9, 11,12	1,2,3, 4
Assessment 5: Final Exam		50%		1, 2, 3, 4

There are **3 hurdle** requirements on the final exam.

- Hurdle requirement **#1:** in the final exam you must solve a task by writing a program that uses an **array**.
- Hurdle Requirement **#2:** in the final exam you must solve a task by writing a program that uses a **linked list**.
- Hurdle Requirement **#3:** in the final exam, you must score **at least 40%**

You **cannot** pass CP1511 unless you achieve **ALL** of the above hurdles.

Final Mark

Your final mark for this course will be computed using the above assessments as follows:

CourseWorkMark	=	PracticalExamMark + LabMark + Ass1Mark + Ass2Mark	out of 50
ExamMark	=	ExamPracMark + ExamTheoryMark	out of 50
ExamOK	=	ExamMark ≥ 22.5/50 && Pass (List Hurdle & Array Hurdle)	true/false
FinalMark	=	CourseWorkMark + ExamMark	out of 100
FinalGrade	=	UF, if !ExamOK && FinalMark ≥ 50 FL, if FinalMark < 50/100 PS, if 50/100 ≤ FinalMark < 65/100 CR, if 65/100 ≤ FinalMark < 75/100 DN, if 75/100 ≤ FinalMark < 85/100 HD, if FinalMark ≥ 85/100	

5.2 Assessment criteria and standards

In all programming work, the primary assessment criterion is correctness (i.e. does the code produce the expected output/behaviour according to the exercise specification). This will be tested by executing code against a variety of test cases, some of which are available to students, and others of which are used after submission for assessment purposes. Code is also expected to be expressed clearly, with consistent formatting and using relevant variable names.

5.3 Submission of assessment tasks

All assignments will be submitted online via CSE’s submission system. Late assignments submissions will be penalized. The exact penalty will be specified in the assignment specification - typically it is 2% reduction in maximum mark for every hour late.

If you are unable to submit an assignment by the due date, due to medical reasons or other reasons which significantly affect your ability to carry out your work, you should contact the lecturer as soon as possible, preferably well before the assignment deadline. If the lecturer considers that your ability to complete the assignment on time has been adversely affected, an extension may be granted to make up for the time you were unable to work on the assignment.

Lab exercises must be submitted by the specified due date of the exercises, likely on the Sunday.

5.4 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.

Plagiarism is defined as using the words or ideas of others and presenting them as your own. **UNSW treats 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
- UNSW Plagiarism Procedure

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

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

- Student Code Policy
- Student Misconduct Procedure
- Plagiarism Policy Statement
- Plagiarism Procedure

5.4. Feedback on assessment

Assignments will be marked after the submission deadline and annotated with comments by the tutor. You can discuss the tutor's comments in a lab class after you have received the feedback.

You may seek feedback from your lab demonstrators for your lab submission during the lab class in the week following the submission.

6. Readings and resources

The course website will make available all lecture material along with code examples.