COMP1511 - Programming Fundamentals

- Convenor/Stream A Lecturer: Andrew Taylor andrewt@unsw.edu.au
- Stream B Lecturer: Andrew Bennett andrew.bennett@unsw.edu.au
- Stream C Lecturer: John Shepherd jas@cse.unsw.edu.au
- Tutors: too many to list - see class web page
- Class webpage: https://cgi.cse.unsw.edu.au/ cs1511/

All course information is placed on the course web site. COMP1511 (and other COMP courses) does not use Moodle.

COMP1511 - Help

- course outline (linked to class webpage)
- lecture recordings (linked to class webpage)
- help sessions (listed on class webpage)
- talk to Andrew/Andrew/John immediately after a lecture
- talk to your tutor
- course forum (linked to class webpage)
- extraordinary matters make an appointment with Andrew Taylor (`andrewt@unsw.edu.au`)
- CSE Student Office (K17 G04) for enrollment/course/academic issues
- CSE Help Desk for system problems `http://www.cse.unsw.edu.au/~helpdesk/`

About COMP1511

- introductory programming course
- no prerequisites
- assumes zero previous programming experience
- fundamental programming concepts
- solve problems with C programs
- problem solving - design, testing, debugging

COMP1511 vs COMP1911

COMP1511 & COMP1911 assume no programming experience.

CS majors must take COMP1511.

Non-CS majors with an interest in coding/CS should take COMP1511

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

Many COMP courses effectively require COMP1511
We also offer a mid-year bridging course for student who take COMP1911 and discover they should have taken COMP1511.

Lectures

- Monday 13:00 - 15:00
- Wednesday 14:00 - 16:00

1000+ students requires three lecture streams.

- Stream A - Central Lecture Block 7
- Stream B - Mathews Theatre A
- Stream C - Mathews Theatre B

2nd hour of Wednesday lecture be used for revision or challenge material in later weeks.
Feel free to ask questions, but otherwise *quiet* so others can hear.
Streams will cover same content each lecture, but take slightly different approach.
Recordings & slides from streams will be posted to class web page.
Stream C may merge into other streams when room in A theatres.

Lectures

Lectures will:

- present a brief overview of theory
- focus on practical demonstrations of coding
- demonstrate problem-solving (testing, debugging)

Lecture slides available on the web before lecture.
Feel free to ask questions, but otherwise *quiet please*.
Lectures recorded and linked to home page.

Tutorials

Tutorials aim to:

- clarify any problems with lecture material
- work through problems related to lecture topics
- give practice with design skills *(think before coding)*

Tutorials and labs start in week 1.
Tutorial questions available on the web the week before.
Tutorial answers available on the web after the week's last tutorial.
Use tutorials to discuss *how* solutions were reached.

Tutorials

Attempt the problems yourself beforehand
Your tutor may ask for your attempt to start a discussion.
Do *not* keep quiet in tutorials ... talk, discuss, ...
Don't let you tutor go too fast (interact!)
Extra tute questions each week for revision.

Lab Classes

Each tutorial is followed by a two-hour lab class.

- Lab exercises mostly small coding tasks.
- Lab exercise build skills need for assignments & exam.
- Lab exercises done in pairs.
- Tutors will form pairs and reorganize them every 4 weeks.
- Often include challenge exercises.
- Challenge exercises may be individual (not pair)
- Both members of pair must submit with **give**
- automarked (with partial marks)
- 12% of final mark.
- full marks possible without completing all challenge exercises

Weekly Programming Tests

- programming tests weeks 3-12
- immediate reality-check on your progress.
- done in your own time under self-enforced exam conditions.
- time limit of 1 hour
- automarked (with partial marks)
- contribute 8% of final mark.
- best 8 of 10 tests used to calculate the 8
- any violation of the test conditions, zero for whole component

Assignments

Assignments give you experience with larger programming problems than the lab exercises

Assignments will be carried out individually.

They always take longer than you expect.

Don't leave them to the last minute.

There are late penalties applied to maximum assignment marks, typically 2%/hour

Code of Conduct

COMP1511 will offer inclusive learning environment for all atudents.
In anything conttected to COMP1511 including social media, these things are student misconduct and will not be tolerated

- racist/sexist/offensive language or images
- sexually inappropriate behaviour
- bullying, harrassing or aggressive behaviour
- invasion of privacy

Show respect to your fellow students and the course staff

Plagiarism

What is plagiarism?

Presenting the (thoughts or) work of another as your own.

Cheating of any kind constitutes academic misconduct and carries a range of penalties. Please read course intro for details.

Examples of inappropriate conduct:

- groupwork on individual assignments (discussion OK)
- allowing another student to copy your work
- getting your hacker cousin to code for you
- purchasing a solution to the assignment

Plagiarism

What is plagiarism?

Presenting the (thoughts or) work of another as your own.

Cheating of any kind constitutes academic misconduct and carries a range of penalties. Please read course intro for details.

Examples of inappropriate conduct:

- groupwork on individual assignments (discussion OK)
- allowing another student to copy your work
- getting your hacker cousin to code for you
- purchasing a solution to the assignment

Remember

You are only cheating yourself and chances are you will get caught!

Plagiarism

Labs must be entirely the work of your pair.
Assignments must be entirely your own work.
You can not work on assignment as a pair (or group).
Plagiarism will be checked for and *penalized*.
Plagiarism may result in suspension from UNSW .
Scholarship students may lose scholarship.
International students may lose visa.
Supplying your work to any another person may result in loss of all your marks for the lab/assignment.

Exam

Held in the CSE Labs (must know lab environment)

Format:

- on-line documentation
- mostly we give you tasks
- you write C program to solve them
- also may ask you to read C code or other written question

How to pass the Exams

- do the lab exercises
- do the assignments *yourself*
- practise programming outside classes
- treat extra tutorial questions like a mini prac exam

Assessment

- 12% Labs
- 8% Weekly Programming Tests
- 6% Assignment 1 - due week 8
- 12% Assignment 2 - due week 12
- 12% Assignment 3 - due week 12
- 50% Final Exam

Any of the above marks may be scaled to ensure an appropriate distribution, and to ensure consistency across exam sessions. Typically scaling is not required.

Hurdle Requirements

To pass the course, you must do all of these:

- score 50/100 overall
- solve problem using arrays in final exam
- solve problem using linked-lists in final exam

Supplementary Assessment

Students will be offered a supplementary exam if they miss the original exam due to (documented) illness or misadventure.
Also automatic supp if your mark is 40-49 and you attended 9+ labs, did 8+ programming tests and attempted all assignments.
Also automatic supp if your mark is 50+ but you fail the hurdle.
The supp tentatively scheduled for Thursday 26 July
Your responsibility to be available - no alternative!

Course Text

Optional Course text

Programming, Problem Solving, and Abstraction with C
Alistair Moffat, Pearson Educational, Australia, 2012,
ISBN 1486010970

- good textbook - recommended if you want a text
- not required

Email

UNSW students are automatically given a zmail address.

It looks like: `z1234567@@unsw.edu.au` or `d.ritchie@unsw.edu.au`

You must read it, important information is sent to it.

If you redirect your zmail address, e.g. to dmr@gmail.com - test the forwarding!

You should already have received a welcome COMP1511 e-mail

How to succeed in COMP1511

Successful COMP1511 students:

- prepare for tutorials and participate
- work on lab exercises before and after labs
- start assignments early
- do assignments and labs themselves
- practice - code, code, code
- don't panic - think, persevere