COMP1511 - Introduction to Programming

- Lecturer/Convenor: Andrew Taylor andrewt@unsw.edu.au
- Stream B Lecturer: John Shepherd jas@cse.unsw.edu.au

Consultation times posted on the class web page.

For extraordinary matters make an appointment with Andrew.

Most matters talk to Andrew/John immediately after a lecture or by posting a question on the class forum.

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.
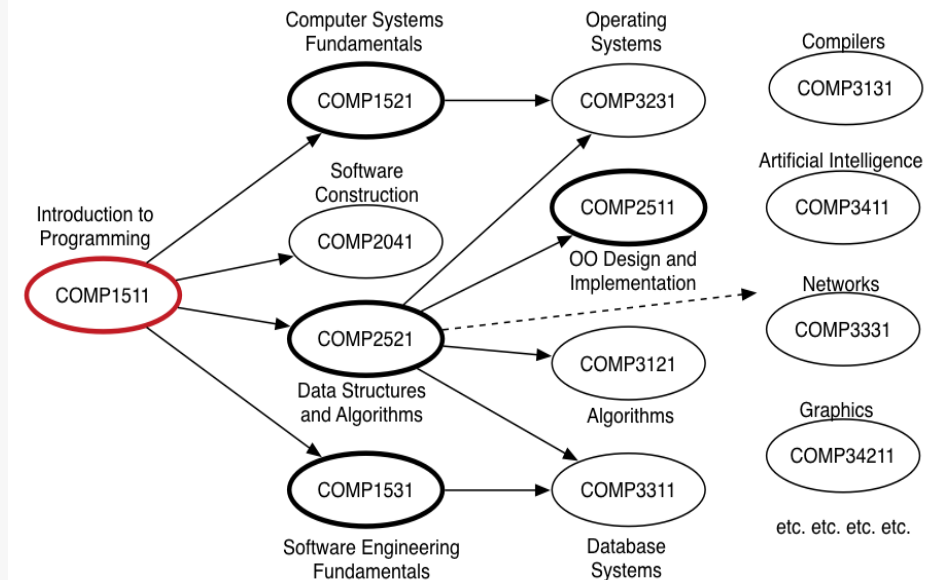
CS majors must take COMP1511.

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.

COMP Courses

Lectures

900+ students requires two lecture streams.

Stream A - lecturer Andrew Taylor
Monday 11:00 - 13:00 Keith Burrows Theatre
Wednesday 11:00 - 13:00 Science Theatre

Stream A - lecturer John Shepherd
Monday 11:00 - 13:00 Mathews Theatre A
Wednesday 11:00 - 13:00 Mathews Theatre A

2nd hour of Wednesday lecture be used for revision or challenge
material in later weeks.

Feel free to ask questions, but otherwise *quiet please*.
Lectures recorded and linked to home page.

Lectures

Lectures will:

- present a brief overview of theory
- focus on oractical 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 week1.
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 aim to build skills that will help you to

- complete the assignment work
- pass the final exam

Lab classes give you experience applying tools/techniques.
Each lab exercise is a small implementation/analysis task.
Lab exercises done in pairs.
Tutors will form pairs and reorganize them every 4 weeks.
Often includes a challenge exercise needed to get an A+
Do them yourselves!   and   *Don't fall behind!*

Lab Classes

Lab exercises contribute 10% to overall mark.
In order to get a mark, lab exercise for Week X must be

- submitted via `give` before Monday 11:00 at end of week X
- demonstrated to the tutor *during* the Week X lab
  OR, demonstrated *at the start of* the Week X+1 lab

There are more than 10 marks available for labs.
The first lab in week 1 is designed to help you familiarise yourself
with the CSE Linux lab environemnt.

Assignments

Assignments give you experience applying tools/techniques
(but to 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

Organising your time

$$\Rightarrow$$

no penalty.

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.

Exams

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 turorial questions like a mini prac exam

Assessment

- 10% Labs
- 5% Blog
- 15% Assignment 1 - due week 8
- 10% Exam #1 (Arrays) week 8/9
- 15% Assignment 2 - due week 12
- 10% Exam #2 (Linked Lists) week 13
- 35% 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 either week 8 or final exam
- solve problem using linked-lists in either week 13 or 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 have attended 9+ labs and reasonable attempts on both assignments + blog.
Also automatic supp if your mark is 50+ but you fail the hurdle.
The supp tentatively scheduled for Thursday 20 July
Your responsibility to be available - no alternative!

Course Website

All course information is placed on the course web site.

- `http://www.cse.unsw.edu.au/~cs1511/`

Most UNSW courses use Moodle.
COMP1511 (and other COMP courses) does not use Moodle.

Other Sources of Information/Help

- course outline (linked to class webpage)
- lecture recordings (linked to class webpage)
- class consultations (listed on class webpage)
- your tutor
- course forum (linked to class webpage by Thursdaysoon)
- Andrew/John: after lectures, by e-mail
  `andrewt@unsw.edu.au` `jas@cse.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/`

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

Todo ASAP

- Find course website:
  `http://www.cse.unsw.edu.au/~cs1511/`
- Read course outline.

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