COMP1100: Programming as Problem Solving

Lecture 1: Introduction

Dr. Liam O'Connor

(based on material from Ranald Clouston, Yun Kuen Cheung, and Michael Norrish)

School of Computing, Australian National University

Semester 2 2025

Acknowledgement of Country

We acknowledge and celebrate the Traditional Owners of the land on which we meet, the Ngunnawal and Ngambri peoples, and also the Traditional Owners of Country throughout Australia.

We pay our respects to Elders past, present, and emerging.

Learn more about Acknowledgement of Country.

From my photo of “Ngunnawal Land”, by Eddie Longford

Teaching Team

Dr. Liam O'Connor

Lectures from Week 1 to Week 5

Dr. Ranald Clouston

Lectures from Week 6 to Week 12

Tutors for Lab Sessions

- Liz Yevdokimov
- Thomas Fahey
- Sophie Wegener
- Georgia Donoghue
- Malcolm Macdonald

Check MyTimetable to see your lab times or to change labs.

Attend three 1hr **lectures** per week and one 2hr **lab** per week.

Drop-in sessions are TBA.

This Course (COMP1100)

This is the entry point for study about Computer Science, using the Haskell programming language.

If you just want to do one programming course alongside your non-computing degree, consider **COMP1730** (Python).

You *don't* need

Any prior programming or computing background.

You *do* need

to be comfortable with mathematical thinking. If you struggle, take **MATH1005**.

COMP1130 does not run in Semester 2

Computer Science

The word “computer” was originally a job title.

They were provided an algorithm, presented in natural language (e.g. English) or mathematics, along with data.
They carried out a computation!

Photo c/o Early Office Museum Archives

Computer Science

The word "computer" was originally a job title.

They were provided an algorithm, presented in natural language (e.g. English) or mathematics, along with data.
They carried out a computation!

Dorothy Vaughan (1910-2008)

- Trained as a mathematician, worked as a teacher.
- Joined Langley Memorial Aeronautical Laboratory (later NASA) in 1943 as a computer.
- Rose to head of West Area Computing by 1949, making important contributions to data analysis for space travel.
- Reinvented herself as a programmer for (electronic) computers, using FORTRAN in the 1960s.

See Margot Lee Shetterly, "Hidden Figures" (2016) — also a film!

Computer Science

The word "computer" was originally a job title.

They were provided an algorithm, presented in natural language (e.g. English) or mathematics, along with data.
They carried out a computation!

vs. Electronic Computers

Electronic computers are faster and more accurate, but they have no common sense, no domain knowledge, and no understanding of natural language.

Dorothy Vaughan (1910-2008)

- Trained as a mathematician, worked as a teacher.
- Joined Langley Memorial Aeronautical Laboratory (later NASA) in 1943 as a computer.
- Rose to head of West Area Computing by 1949, making important contributions to data analysis for space travel.
- Reinvented herself as a programmer for (electronic) computers, using FORTRAN in the 1960s.

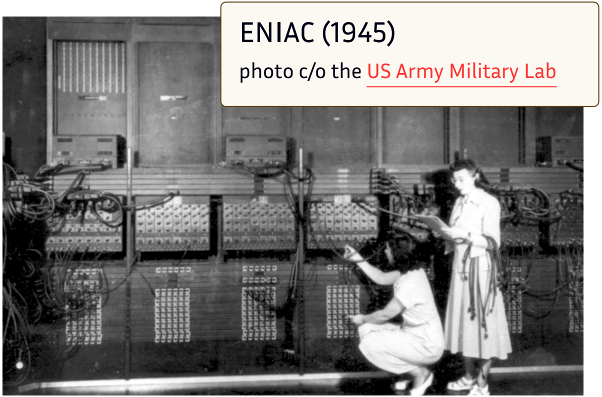
See Margot Lee Shetterly, "Hidden Figures" (2016) — also a film!

The First Programmers

Ada Lovelace (1815-1852)

- In her "Notes", wrote the first ever *computer programs*.
- Intended for her friend Charles Babbage's Analytical Engine, a steam-powered computer that he never managed to build.

The First Programmers



ENIAC (1945)

photo c/o the US Army Military Lab

Ada Lovelace (1815-1852)

- In her "Notes", wrote the first ever *computer programs*.
- Intended for her friend Charles Babbage's Analytical Engine, a steam-powered computer that he never managed to build.

First working programmers: Kathleen McNulty, Frances Bilas, Betty Jean Jennings, Ruth Lichtermann, Elizabeth Snyder, Marlyn Wescoff.

See: Jennifer S. Light, "When computers were women.", Technology and Culture (1999)

Low-level to High-level

```
0x100000e9b   c7 45 f0 00 00 00 00
0x100000ea2   c7 45 ec 00 00 00 00
0x100000eb9   8b 45 ec
0x100000ebc   3b 45 f4
0x100000ebf   0f 8d 21 00 00 00
0x100000ec5   48 63 45 ec
0x100000ec9   48 8b 4d f8
0x100000ecd   8b 14 81
0x100000ed0   03 55 f0
0x100000ed3   89 55 f0
0x100000ed6   8b 45 ec
0x100000ed9   05 01 00 00 00
0x100000ede   89 45 ec
0x100000ee1   e9 d3 ff ff ff
0x100000ee6   8b 45 f0
0x100000ee9   99
0x100000eea   f7 7d f4
```

Machine Code

This is the “native language” of the CPU.

- Nearly incomprehensible to humans.
- Different CPU’s speak different “languages”, so machine code is not portable.

Low-level to High-level

```
0x100000e9b   movl $0x0, -0x10(%rbp)
0x100000ea2   movl $0x0, -0x14(%rbp)
0x100000eb9   movl -0x14(%rbp), %eax
0x100000ebc   cmpl -0xc(%rbp), %eax
0x100000ebf   jge 0x100000ee6
0x100000ec5   movslq -0x14(%rbp), %rax
0x100000ec9   movq -0x8(%rbp), %rcx
0x100000ecd   movl (%rcx,%rax,4), %edx
0x100000ed0   addl -0x10(%rbp), %edx
0x100000ed3   movl %edx, -0x10(%rbp)
0x100000ed6   movl -0x14(%rbp), %eax
0x100000ed9   addl $0x1, %eax
0x100000ede   movl %eax, -0x14(%rbp)
0x100000ee1   jmp 0x100000eb9
0x100000ee6   movl -0x10(%rbp), %eax
0x100000ee9   cltd
0x100000eea   idivl -0xc(%rbp)
```

Assembly Language

A (barely) human-readable language which gives names to each machine instruction. It gives high control to the programmer, but:

- It is no more portable than machine code.
- It is still very hard and error-prone to write programs in.
- Our thinking is constrained by the language of the machine.

Low-level to High-level

```
int sum = 0;
int i = 0;
loop:
  if (i >= length(values))
     goto done;
   sum += values[i];
   i = i + 1;
   goto loop;
done:
   mean = sum / i;
```

Higher-level programming languages

- More readable to humans.
- Can be portable.
- Our thinking is less constrained by the language of the machine.

Source code (the program text) must be transformed into machine code:

Interpreter Executes code line-by-line.

Compiler Transforms whole program.

Grace Hopper, Barbara Liskov, Frances Allen

Computing Science

```
mean values
   = sum values / length values
```

Haskell

In this course, we use the very high-level **Haskell** programming language.

We don't want to be constrained by the machine — Computer science is about computing, not computers.

Computing Science

```
mean values
   = sum values / length values
```

Haskell

In this course, we use the very high-level **Haskell** programming language.

We don't want to be constrained by the machine — Computer science is about computing, not computers.

Ada Lovelace

"The *science of operations*, as derived from mathematics more especially, is a science of itself, and has its own abstract truth and value"

E. W. Dijkstra

"Computer science is no more about computers than astronomy is about telescopes."

Computer *Science*

Computer *Science*

Scientists do Experiments!

If you're curious about something:

Try it and see!

Computer *Science*

Scientists do Experiments!

If you're curious about something:

Try it and see!

Scientists use Mathematics!

We will use mathematical principles to understand our programs.

Computer *Science*

Scientists do Experiments!

If you're curious about something:

Try it and see!

Scientists use Mathematics!

We will use mathematical principles to understand our programs.

Scientists observe Lab Safety!

Look after your (mental and physical) **health**.

- Posture, eye health, sleep patterns.
- If you're struggling, ANU services exist.

Computer *Science*

Scientists do Experiments!

If you're curious about something:

Try it and see!

Scientists use Mathematics!

We will use mathematical principles to understand our programs.

Scientists observe Lab Safety!

Look after your (mental and physical) **health**.

- Posture, eye health, sleep patterns.
- If you're struggling, ANU services exist.

Scientists work Collaboratively!

We **help each other** to learn.

But: we don't take credit for other's work.

Whenever you're using others' work, you **must** cite it.

You cannot be awarded marks for work you didn't do yourself.

AI outputs are also **not your work**.

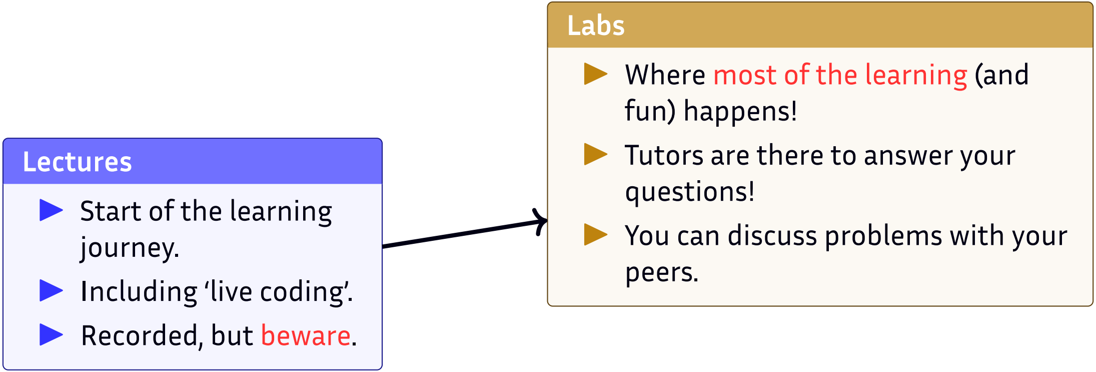
Academic misconduct: We can find out, we will find out.

How to Learn in this Course

Lectures

- Start of the learning journey.
- Including ‘live coding’.
- Recorded, but beware.

How to Learn in this Course



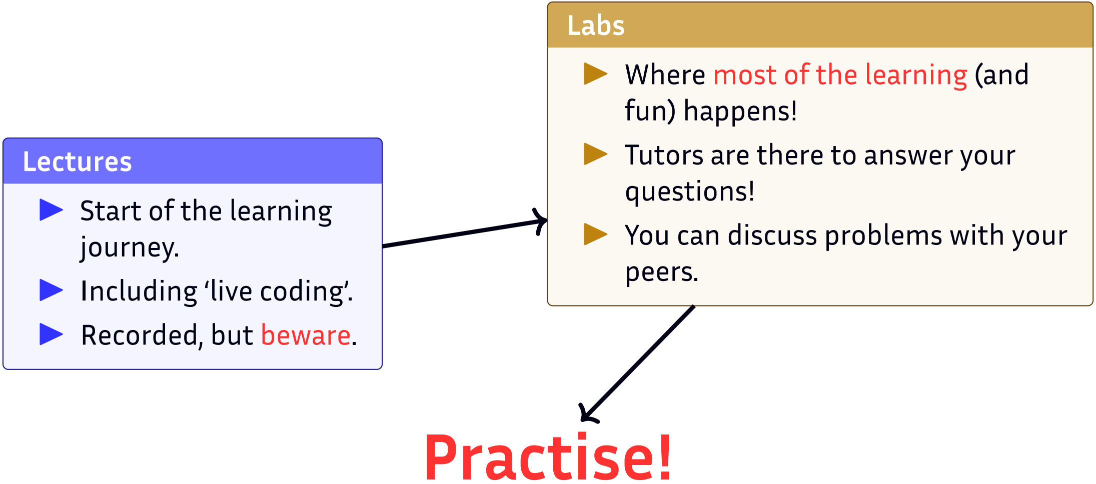
Lectures

- Start of the learning journey.
- Including ‘live coding’.
- Recorded, but beware.

Labs

- Where most of the learning (and fun) happens!
- Tutors are there to answer your questions!
- You can discuss problems with your peers.

How to Learn in this Course



Lectures

- Start of the learning journey.
- Including 'live coding'.
- Recorded, but beware.

Labs

- Where most of the learning (and fun) happens!
- Tutors are there to answer your questions!
- You can discuss problems with your peers.

Practise!

Websites

Wattle and Canvas

- We **do not** use Canvas in COMP1100 (yet)
- We **do** have a Wattle page, but we don't use it much.

Ed

We use Ed, which contains:

- A discussion forum, where lecturers (and students!) will answer your questions.
- **Ed Lessons**, an online platform where you can practise programming in Haskell using only a browser.

This means that you do not need to install Haskell to your computer.

All course content (lecture slides, exercises, labs etc.) **is available from Ed Lessons.**

Websites

Wattle and Canvas

- We
- We

Signup

You should have received an invite email to register for Ed.

Use your ANU uid email
(e.g. `u9876543@anu.edu.au`) to register

Any non-ANU uid emails enrolled into our Ed course will be unenrolled immediately without notification.

Ed

We use

- A di estions.
- **Ed l** Haskell usir

This means that you do not need to install Haskell to your computer.

All course content (lecture slides, exercises, labs etc.) **is available from Ed Lessons.**

Practising Skills Beyond Labs

Practise your Skills!

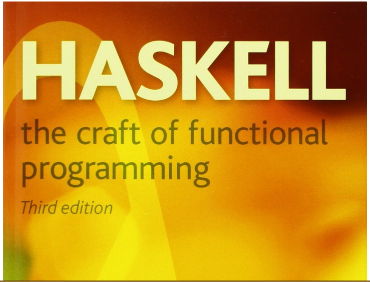
This is the **only way** to become a good programmer.

- **You** must **write** the code.
- Reading a worked solution is not enough.

Other Resources!

- Challenge exercises available on Ed Lessons.
- Past exam papers can be found on links from Ed.

  (**Warning**: COMP1100 changes over time!)



Textbook

Haskell: the Craft of Functional Programming, by Simon Thompson.

- Available online for free.
- Copies can be purchased or loaned from the library.
- **Many** exercises within.

Getting Questions Answered

In-class

- Live questions in lectures are welcome!
- Your tutor is there to answer questions in labs.

  (if they relate to the current content)

Getting Questions Answered

In-class

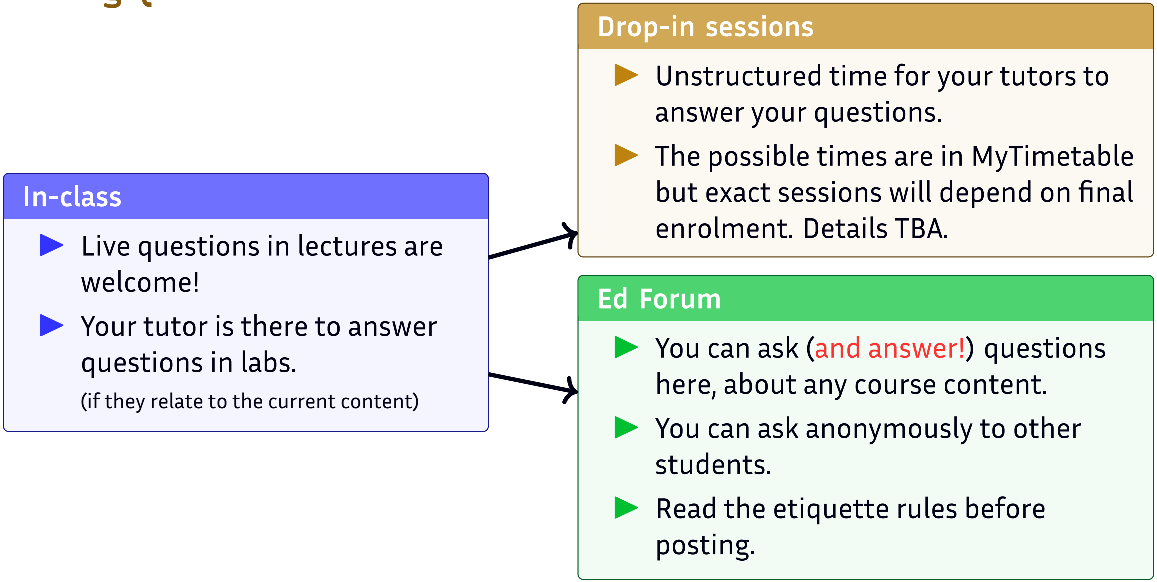
- Live questions in lectures are welcome!
- Your tutor is there to answer questions in labs.

  (if they relate to the current content)

Drop-in sessions

- Unstructured time for your tutors to answer your questions.
- The possible times are in MyTimetable but exact sessions will depend on final enrolment. Details TBA.

Getting Questions Answered



In-class

- Live questions in lectures are welcome!
- Your tutor is there to answer questions in labs.

  (if they relate to the current content)

Drop-in sessions

- Unstructured time for your tutors to answer your questions.
- The possible times are in MyTimetable but exact sessions will depend on final enrolment. Details TBA.

Ed Forum

- You can ask (and answer!) questions here, about any course content.
- You can ask anonymously to other students.
- Read the etiquette rules before posting.

Key Dates

Monday 25th August

Mid-semester Test, conducted in computer labs at 9:00am.

Sunday 31st August

Semester 2 Census Date: the last day where you can withdraw from courses without an academic or financial penalty.

Friday 3rd October

The last day where you can withdraw from courses without academic penalty.

CSS Student Services can help you if you have any questions regarding courses, enrolments, visas etc.

Email

Does it need to be an email?

In general, questions about course content should be posted to Ed. You can ask questions there that are only seen by conveners and tutors.

When should I email my tutor?

Email your tutor *only* if you have questions specific to your interactions with that tutor (about lab attendance, assignment marking etc.)

When should I email the lecturers?

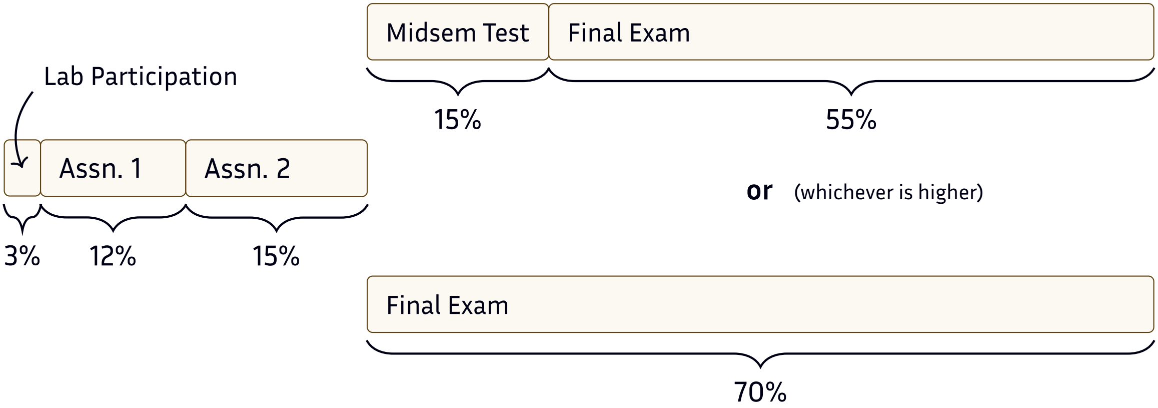
Email the lecturers, instead of using Ed, *only* if you have an issue that tutors could not help with or should not see.

- Use `comp1100@anu.edu.au` (Liam and Ranald) only
- Use your ANU address

All other enquiries

Any enquiries not directly related to this course should be directed to CSS Student Services.

Assessment



Lab Participation

Assn. 1

Assn. 2

3%

12%

15%

Midsem Test

Final Exam

15%

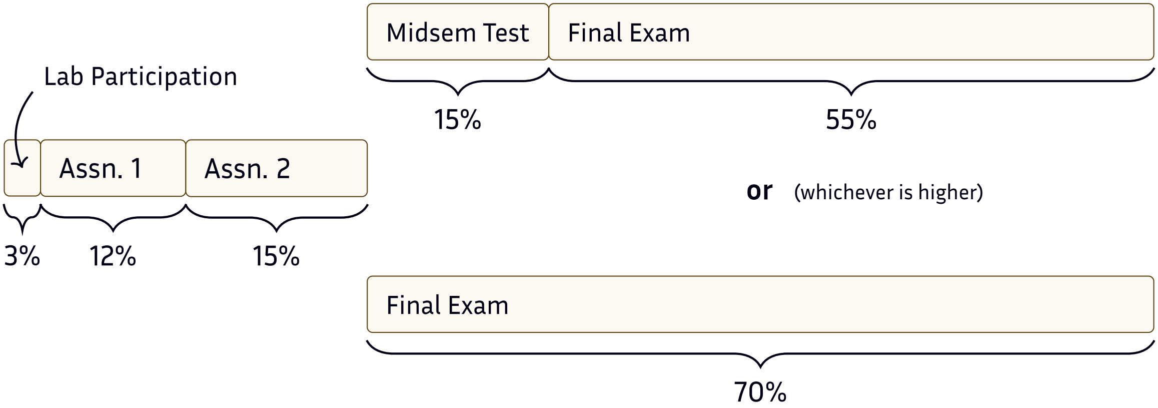
55%

or (whichever is higher)

Final Exam

70%

Assessment



Lab Participation

Assn. 1

Assn. 2

3%

12%

15%

Midsem Test

Final Exam

15%

55%

or (whichever is higher)

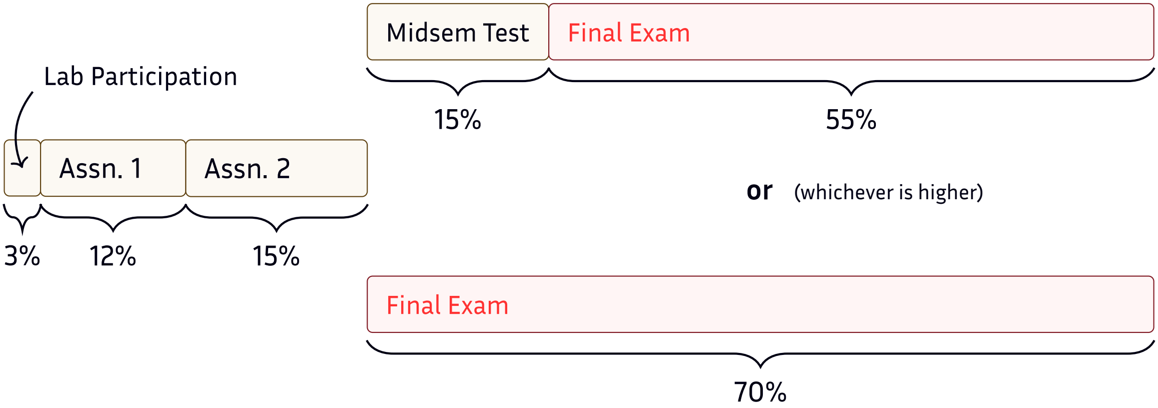
Final Exam

70%

Course Grade Requirement

To pass this course you must get $\geqslant$ 50% of the above assessment marks.

Assessment



Lab Participation

Assn. 1

Assn. 2

3% 12% 15%

Midsem Test

Final Exam

15% 55%

or (whichever is higher)

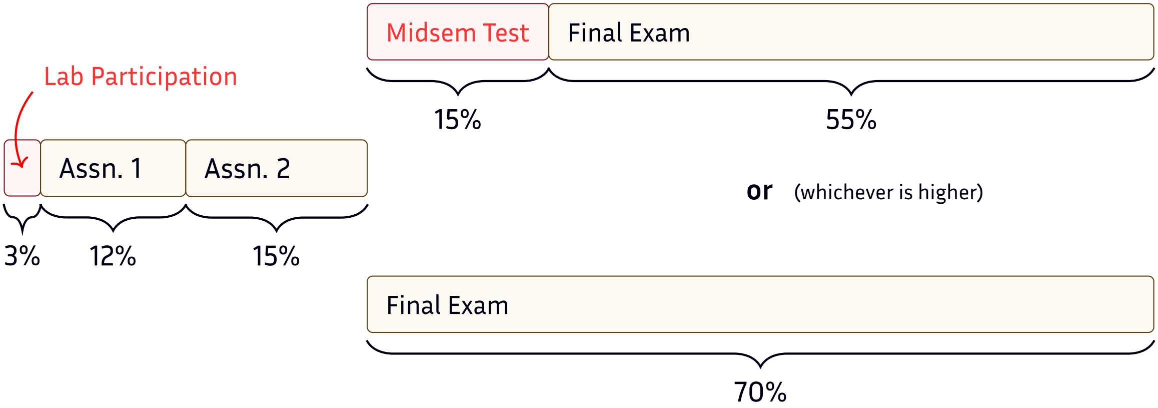
Final Exam

70%

Exam Hurdle

To pass this course you must also get $\geqslant$ 40% of the marks in the Final Exam.

Assessment



Lab Participation

Assn. 1

Assn. 2

3% 12% 15%

Midsem Test

Final Exam

15% 55%

or (whichever is higher)

Final Exam

70%

Engagement Hurdle

To pass this course you must also participate in 4 of the first 5 labs **or** get ⩾ 30% of the mid-semester test.

Academic Misconduct

What Counts as Plagiarism

- Work is considered not your own if it looks so much like some other work (e.g., that of another student) that we cannot believe that the work was produced individually.
- Viewing another person's assignment code, or report text, is **extremely dangerous**. But it is possible to plagiarise without seeing each other's code, if discussions about assignments become overly detailed.
- Any use of sources without citation. But it is still plagiarism if you use another's solution for an individual assignment, even with citation.

Academic Misconduct

What Counts as Plagiarism

- Work is considered not your own if it looks so much like some other work (e.g., that of another student) that we cannot believe that the work was produced individually.
- Viewing another person's assignment code, or report text, is **extremely dangerous**. But it is possible to plagiarise without seeing each other's code, if discussions about assignments become overly detailed.
- Any use of sources without citation. But it is still plagiarism if you use another's solution for an individual assignment, even with citation.

What Doesn't Count as Plagiarism

- Discussions (e.g about assignments) that remain at the level of concepts but not solutions.

  For example, discussions of relevant parts of the lecture notes.
- Anything to do with lab material, where group study is fine.

  But we do not recommend getting someone else to do your labs!

Academic Misconduct

What Counts as Plagiarism

- Work is considered not your own if it looks so much like some other work (e.g., that of another student) that we cannot believe that the work was produced individually.
- Viewing another person's assignment code, or report text, is **extremely dangerous**. But it is possible to plagiarise without seeing each other's code, if discussions about assignments become overly detailed.
- Any use of sources without citation. But it is still plagiarism if you use another's solution for an individual assignment, even with citation.

What Doesn't Count as Plagiarism

- Discussions (e.g about assignments) that remain at the level of concepts but not solutions.

  For example, discussions of relevant parts of the lecture notes.
- Anything to do with lab material, where group study is fine.

  But we do not recommend getting someone else to do your labs!

73% of the course is not assignments — so plagiarists will fall short anyway.

Academic Misconduct

What Happens for Plagiarism Cases

- There is a stressful and sometimes prolonged investigation, including an interview.

  Please be civil if you are asked to participate in such an interview
- Assessment marks are reduced heavily, usually to 0%
- Information is recorded against your name by the university, which will be seen by academics in any setting where the (internal) academic transcript is relevant.

Academic Misconduct

What Happens for Plagiarism Cases

- There is a stressful and sometimes prolonged investigation, including an interview.

  Please be civil if you are asked to participate in such an interview
- Assessment marks are reduced heavily, usually to 0%
- Information is recorded against your name by the university, which will be seen by academics in any setting where the (internal) academic transcript is relevant.

A Warning about 'Tutors'

Approach people or organisations offering 'tutoring' with skepticism:

- Some of these people and organisations are plagiarism facilitators.
- Some of them have even engaged in blackmail.

Note that we **regularly** catch people who use these 'services'. Just because they are expensive, doesn't mean they care if their clients get caught.

Class Representatives

Class Representatives

- Class Representatives are a core part of ‘student partnership’, a principle to which the ANU has committed.
- A Class Representative is the official liaison between students and the course convener.
- The primary responsibility of a Class Representative is to provide constructive feedback on issues affecting the entire student cohort, or a subset of the student cohort.
- Class representatives also need to attend regular meetings arranged by School and College to provide reports on feedback.

Benefits of Being a Class Rep

- Ensure students have a voice to their course convener, lecturer, tutors, and College.
- Develop skills sought by employers, such as interpersonal, dispute resolution, leadership and communication skills.
- Become empowered, play an active role in determining the direction of your education.
- Become more aware of issues influencing your University and current issues in higher education.

Class Representatives

Class Representatives

- Class Representatives are a core part of 'st th
- A lia co
- Th Re fe st co
- Cl regular meetings arranged by School and College to provide reports on feedback.

Benefits of Being a Class Rep

- Ensure students have a voice to their and
- ers,
- ve role our
- urrent issues in higher education.

Call for Class Reps

We are looking for 2 class representatives!

- Looking for diverse representation – male and female, Australian and international, different degrees etc.
- If you are interested, please submit your application to `comp1100@anu.edu.au` before Monday week 2.
- Briefly write why you might be a good class representative. Describe your background or anything that might be relevant.
- We will announce the class representatives in week 2 or 3.

Fifty50

FIRST YEAR MENTORING PROGRAM

Are you a **first year STEM student** learning how to navigate uni life?

Or a **2nd+ year STEM student** keen to share your hard-earned wisdom and uni tips?

Sign up as a **mentor** or a **mentee** below!