

# ENGG1811: Computing For Engineers

## Introduction

Week 1: Tuesday 16<sup>th</sup> September, 2025

Monday 14:00 - 16:00 | HarpM15570

# We will get started soon!

- ▶ We will start 5 minutes past the hour.
- ▶ We'll begin with an icebreaker: introduce one of the people sitting next to you in the lab.
  - ▶ So please start getting to know someone near you now.
  - ▶ See the *Now Your Turn: Introductions* slide for some icebreaker questions. If you are viewing this live in the lab on the screen, you can also access these slides through the email I sent out.

# Today

General Course Information & Expectations

Arithmetic Operations in Spyder & Python

Lab Tips

## General Course Information & Expectations

# Introductions



**Brendan Mabbutt**

[b.mabbutt@student.unsw.edu.au](mailto:b.mabbutt@student.unsw.edu.au)

**Degree:** Advanced Mathematics

**Likes:** Snow, Uni Holidays, Programming

**Dislikes:** Winter, Public Holidays, Math



**Jenny Ji**

**Degree:** Material Science and Engineering

**Likes:** Sleep

**Dislikes:** Running and Studying

## Now your turn: Introductions

Some icebreakers I'll ask, try to answer some of them for someone next to you:

- ▶ When is their name?
- ▶ What is their degree?
- ▶ What are their likes and dislikes?

Be sure to get to know the people around you during the lab — it's an important part of learning in this course!

# Emails

- ▶ Has everyone received the welcome email ?
- ▶ Ensure that your UNSW student email (Outlook) is activated.
  - ▶ If you haven't set it up yet, visit: [here](#)
- ▶ Please send me an email if you have any questions related to the labs. If you have more general issues with the course (e.g. self-directed labs or setup), contact the course account at [en1811@cse.unsw.edu.au](mailto:en1811@cse.unsw.edu.au).
- ▶ If you feel there is a situation out of your control affecting your performance, consider applying for special consideration: [here](#). Feel free to loop us in. The same goes for ELPs.

# Why I Dislike Public Holidays

- ▶ There is a public holiday in Week 4 on Monday 6th October ( Labour Day ).
- ▶ The lab will not be held in person.
- ▶ We will reschedule the lab, likely at the same time but online.
- ▶ You may also attend another lab that week in person and get marked there if you prefer. Send me an email and I'll organise this for you. Timetable for alternative labs is [here](#).
- ▶ Please let me know if you have any questions or concerns.
- ▶ I'll release more details later.



# What is ENGG1811?

- ▶ This course serves as an introduction to programming with engineers from the non-electrical disciplines in mind
  - ▶ We will often motivate problems and ideas by leaning on physical examples and 'real'-world systems
  - ▶ You will NOT come-away from this course knowing how to build your own website , or exactly what is going on inside your computer when you program
  - ▶ You will come-away from this course knowing, instead, how to automate calculations and to create meaningful data that is crucial for an engineer

# What will be doing in ENGG1811?

- ▶ The plan is to use Python as our programming language of choice in order to learn common programming and computation techniques
  - ▶ Our IDE will be Spyder
  - ▶ To use spyder at home, use the Anaconda package manager — see the setup guide on the course website.
- ▶ We will also provide a brief introduction to Excel & Matlab

# Assessments

Item	Weighting	Due Date
Labs	20%	Each week in class
Assignment 1	20%	Week 7
Assignment 2	20%	Week 10
Final Exam	40%	Exam Period

The **final exam** will also have a **40 % hurdle**, meaning you will need to get at least 40% on the final exam and at least 50% overall to pass the course.

# Important Resources

- ▶ **ENGG1811 Course Website** : All announcements and resources (labs, help sessions, live coding session, lecture slides, assignments, sample exams, timetable, etc.).
  - ▶ Log in with your zID and zPass (not your email address).
- ▶ **Discourse Forum** : Moderated by tutors for course-specific questions. Students often help each other, and we encourage this!
  - ▶ Join via the course website.
- ▶ **Live-coding session**: Callum will break down optional problems weekly on Tuesdays 1–2PM (starting Week 2).
- ▶ **Help sessions**: One-on-one help with concepts, labs, or assignments. Other students with similar questions may attend and help!

# Labs

- ▶ Our labs will be held each Monday from 14:00 - 16:00.
- ▶ Labs are marked in Weeks 2–5 and 7–10:
  - ▶ 2 marks: completing **AND** explaining your lab solutions
  - ▶ 1 mark: answering an MCQ available *only* during your lab time
  - ▶ You may leave early once your work is marked!
  - ▶ Expect to **explain** your solutions.
- ▶ Two **virtual labs** (Excel & Matlab) can be completed in your own time once released.
- ▶ **Come prepared** : exercises are released on the course website beforehand—read them in advance.

# Lab Extensions

- ▶ At my discretion, I may grant a one-week extension for marking lab work:
  - ▶ Week 6 does not count (e.g. an extension in Week 5 carries to Week 7).
  - ▶ I will generally grant these automatically if you are sick or run out of time, provided it's your first request.
  - ▶ Send me an email if you cannot make a lab.
  - ▶ For absences longer than one week, or if I don't grant an extension, you must apply for special consideration: [here](#).
  - ▶ You may also get work marked in another lab: timetable [here](#).

## Demo: The course website, Moodle and Discourse

- ▶ We will go through the course website and explore its important pages.
- ▶ We will look at how to join and post on the Discourse forum. Specifically:
  - ▶ Check if your question has already been asked to avoid duplicates
  - ▶ Ask questions in a general way so code isn't required — tutors can review code in help sessions
  - ▶ Choose the correct category

# Tips for First-Time Programmers

- ▶ Programming is a skill that builds up incrementally ! It's similar to learning mathematics, a language, or even an instrument.
  - ▶ Please practice each week's material, as it will serve as the foundation for future weeks.
- ▶ Just like any skill , you will experience growing pains — be kind to yourself during those moments.
- ▶ Experiment : there's no harm in recklessly playing around with the console and code.
- ▶ Debugging is a skill in itself — take it seriously and be patient with it.
- ▶ Ask for help when needed, but also enjoy the process of struggling and figuring things out — problem-solving and debugging skills are necessary to build for this course.



# Working Together & AI

- ▶ You are allowed and encouraged to work with others on labs , but you are not permitted to directly copy code. You must understand your code and be able to reproduce it if asked.
- ▶ You are not allowed to work with others on assignments , as they must be completed individually.
- ▶ Using AI for assignments or labs is strictly prohibited .
  - ▶ You risk receiving an automatic zero.
  - ▶ You will not have access to AI during the final exam (remember the **HURDLE**), so relying on AI will prevent you from passing the course.

## Arithmetic Operations in Spyder & Python

## Demo: Spyder IDE

- ▶ Here, we will go through how to access the Spyder IDE on the lab computers. We will cover:
  - ▶ Creating a **directory** for ENGG1811 and a **subdirectory** for each lab using the **file manager**
  - ▶ The **Spyder code editor** and the run button (we won't use it today)
  - ▶ The **Spyder console**, where you can perform single-line calculations and view program output
  - ▶ The **help panel**, where we can see current variable values. We will cover tricks later, such as using the up/down keys for previous commands and **clear** to remove console clutter without deleting data
- ▶ You can use Ctrl + to zoom in and Ctrl - to zoom out in Spyder. This is useful for demonstrating your code — I have bad eyesight :(

## Accustoming Ourselves with Spyder

- ▶ At this stage, we will think of Spyder as a glorified calculator.
- ▶ We can go to the `console` at the bottom right of our IDE and compute expressions such as `2 + 3` , `15 * 3` , and `17 / 4` .
- ▶ For the entirety of today's lab, you can do all exercises in the console.

## (Probably) New Operators

Three 'new' operators you might not have seen before:

- ▶ The exponentiation operator ( `**` ), the double asterisk:
  - ▶ `2 ** 2` gives `4` because  $2^2 = 4$
  - ▶ `3 ** 4` gives `81` because  $3^4 = 81$
- ▶ The modulus operator ( `%` ); the percentage sign calculates the remainder of a division problem:
  - ▶ `4 % 3` gives `1` because  $4 \div 3$  leaves remainder 1
  - ▶ `400 % 365` gives `35` because  $400 \div 365$  leaves remainder 35
- ▶ The integer division operator ( `//` ); the double slash returns the integer portion of a division:
  - ▶ `10 // 3` gives `3` because  $10 \div 3 = 3\frac{1}{3}$
  - ▶ `400 // 365` gives `1` because  $400 \div 365 = 1\frac{35}{365}$

## Optional Exercises

- ▶ We are usually familiar with `^` to mean exponentiation — what is `^` reserved for in Python instead?
- ▶ Play around with `**`, `%`, `//` using negative and decimal numbers to see if they behave in ways that meet your expectations

# Precedence

Operator
()
**
Unary +, -
*, /, %, //
Binary +, -

- ▶ Order of operators is top-down. For example, we evaluate `()` before `*`.
- ▶ **Question:** What is unary `-` as opposed to binary `-`?
- ▶ We must use `()` (parentheses) for grouping in Python. The `{}` and `[]` brackets have different meanings in Python syntax.

# Kahoot Time!

- ▶ I haven't mentioned this yet, but for future labs the typical structure will be: I send slides like this and code in advance for you to look at before the lab.
- ▶ At the start of the lab we will play Kahoot , and I will address any concepts people want to go over in more detail.
- ▶ Otherwise, we will move on to marking you off.
- ▶ Kahoots aren't worth marks but are good practice.
- ▶ This week we will play a Kahoot now, then allow you to work on the lab exercises and "mark" you off. Remember, no actual marks for this week.



## Lab Tips

# Lab Tips

- ▶ Exercise 1

- ▶ Complete these using as few parentheses as possible

- ▶ Exercise 2

- ▶ Work out how many years and *leftover* days are in 1,000,000 minutes
    - ▶ For example, if there are 400 days total, the answer will be 1 year and 35 days
    - ▶ Output the number of whole years on one line and leftover days on another line
    - ▶ If you finish early, also try finding the number of leftover minutes and seconds
  - ▶ The answer is given at the bottom, so please check this before you get "marked off"

## Feedback

Feel free to provide anonymous feedback about the lab!



Feedback Form