

# COMP1730/COMP6730

## Programming for Scientists

### Introduction and Administrative Matters

# Announcements

- \* Labs will be run both on campus and online.
  - Labs do not start until Monday next week.
  - Lab group sign-up opens Thursday 27 February at 12:00pm (noon today).
  - If you are not studying remotely, you may only enrol in an online lab group if you are unable to attend **any** face-to-face groups and have permission from the course convener.
- \* **Read the news forum on wattle**
- \* **Particularly the weekly notice**

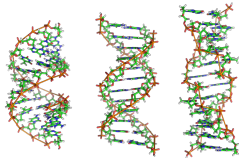


# Lecture outline

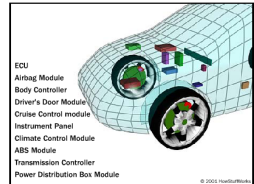
- \* Why learn programming?
- \* Course overview.
- \* Info, contacts & schedule.
- \* Assessment scheme.
- \* Important TODOs.

# Why learn programming?

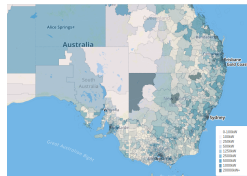
- \* Science rests on data. . . *more and more data.*
  - The Australian SKA Pathfinder radio telescope outputs 2.5GB/s (the SKA is expected to be around 100 times more).
  - A human genome (around 3 billion base pairs) can be sequenced in 3 days.
- \* Processing this data needs software.



- \* Technical systems increasingly run on software.
  - A modern car has over 30 computers running more than 10,000,000 lines of code.



- \* Simulation and optimisation are needed for large-scale design questions.
  - Intermittent renewables accounted for around 8.25% of Australian energy generation in 2017. How do we design the power grid to work with 100%?





*“Whatever branch of engineering you’re in, make sure you know how to program.”*

(Chris Culbert, NASA Chief Technologist)

- \* As a scientist or engineer, you will need to understand how software works, and how to modify or extend it:
  - understand algorithms and implementation to interpret and explain their results;
  - debug programs (find and correct errors);
  - modify existing programs to solve your (unique) problem.
- \* By the end of the course, we hope you'll tackle a novel problem by thinking, “Hey, I can just write a program to solve that. . .”



# Programming example

- \* you want to calculate the monthly cost of a \$300,000 home loan...
  - use one of the on-line calculators?
- \* ...for all loan terms in 10-25 years, and an interest rate of 5.5%, 6.5% or 7.5%.
- \* The formula is

$$A = P \frac{r(1+r)^n}{(1+r)^n - 1}$$

(derive it, or look it up on wikipedia).  
Let's write a program!

```
import math
import matplotlib.pyplot as mpl

def monthly_cost(principal, interest_rate, years):
    monthly_interest_rate = interest_rate/12
    # interest rate is given in % so need to divide by 100
    r = monthly_interest_rate/100
    n_payments = years * 12
    return principal * ((r * math.pow(1 + r, n_payments)) /
                        (math.pow(1 + r, n_payments) - 1))

years = range(10,26)
mc = [monthly_cost(300000, 5.5, y) for y in years]
mpl.plot(years, mc, 'g-')
mc = [monthly_cost(300000, 6.5, y) for y in years]
mpl.plot(years, mc, 'b-')
mc = [monthly_cost(300000, 7.5, y) for y in years]
mpl.plot(years, mc, 'r-')
mpl.show()
```

# Why python?

- \* This is *not* a course on programming in python; it's a course on programming, that uses python.
- \* Python has been consistently ranked in top 5 most popular programming languages,
- \* particularly for science and engineering uses.
- \* Open source, available on most platforms.
- \* Many packages:
  - over 200 in the python standard library;
  - over 100,000 on pypi (`pypi.python.org`).
- \* We will use **python 3**.

# Course description & aims

- \* Introduction to programming (using python).
  - No prior programming or computer science knowledge is required.
  - This does not mean it is easy!
- \* Two aims:
  - Programming as a practical skill.
  - Understand some basic CS concepts; build foundation for later courses.

# Learning outcomes

(revised from ANU Programs & Courses)

Students who succeed in all aspects of this course will:

- \* be able to design and write readable and correct small programs to solve practical data processing problems;
- \* be able to read, understand and debug small computer programs;
- \* understand some practical limitations on computer programs, including scaling (wrt time and memory) and numeric precision (rounding errors) issues.

# Expectations from this Course

- \* This course is not an in depth coverage of everything you can do in Python.
- \* You will not be an expert programmer at the end of this course (unless you are one already).
- \* The focus in this course is on getting you up to speed on how do to something. Please ask me (or your tutor) if you want more information about a particular topic.
- \* Some of the things we talk about in this course will not seem important until your project (or team) gets large.

# Course info & contacts

- \* [cs.anu.edu.au/courses/comp1730/](http://cs.anu.edu.au/courses/comp1730/)
- \* Wattle for forums, quizzes, surveys, assignment submission.
- \* *Read the news & announcements!*
- \* To ask a question:
  - Use the discussion forum on wattle.
  - For *personal* questions, use the course email:  
`comp1730@anu.edu.au`.
  - No regular office contact hours.
  - Consultation sessions every weekday from 4-5pm in HN 1.23.

# Discussion forum – 3 simple rules

## 1. **Read before you post.**

Before posting a question, check if your question has already been answered.

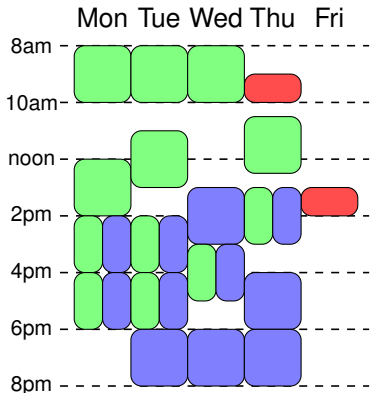
## 2. Give your post a **good, descriptive topic.**

Don't write "A question". Write something like "Variable assignment: why does the value not change?".

## 3. You **may not post** solutions to assignment problems.



# Schedule overview



- \* 2 lectures / week.
- \* 1 2-hour lab / week (from week 2).
- \* See “content & schedule” on course web site.

- \* Except as detailed in the assessment scheme, attendance is never mandatory.

# Assessment scheme (preliminary)

- \* 5 small homework assignments (15%)
- \* 1 larger project assignment (15%)
- \* Mid-semester exam (20%)
- \* Final exam in 1 or 2 parts (50%)

S. Week	
3	Homework 1 due (Thursday)
4	Homework 2 due (Thursday) In lab: Questions on HW 1
5	Homework 3 due (Thursday) In lab: Questions on HW 2
6	Mid-Semester Examination (tbc) In lab: Questions on HW 3
	Break
7	Homework 4 due (Thursday)
8	Homework 5 due (Thursday) In lab: Questions on HW 4
9	In lab: Questions on HW 5
11	Project due (Sunday)
Exam period	Final Examination(s)

- \* The complete assessment scheme is on the course web site at [cs.anu.edu.au/courses/comp1730/assessment](http://cs.anu.edu.au/courses/comp1730/assessment).
- \* The assessment scheme will be final at the end of week 2. Any changes will be announced through the course web page and news forum.
- \* All assignment deadlines are hard – no late submissions will be accepted.
- \* See [www.anu.edu.au/students/program-administration/assessments-exams/](http://www.anu.edu.au/students/program-administration/assessments-exams/) regarding deferred assessments and special consideration.

# Examinations

- \* The course has both a mid-semester and a final examination.
- \* The examinations will take place in the ANU computer labs.
- \* If you defer the mid-semester exam, the mark will be included in the final exam.
- \* You must sit the final examination in person at ANU - but the timing can be delayed until you are here.
- \* The final examination is a hurdle - you must score at least 40% in it to pass the course.

# Academic Honesty

- \* All course assessment is *individual*. You must write your own code, and understand every aspect of what you have written.
- \* Please make sure you are familiar with ANU's academic honesty policy  
<https://www.anu.edu.au/students/academic-skills/academic-integrity>
- \* Any academic misconduct will leave a record on your student file that will never go away.
- \* If you are unsure, please ask your tutor (or me).

# Remote Students

- \* All course material is available online.
- \* There are 11 online lab groups - and remote students have priority for places.
- \* Online labs will be conducted using Microsoft Teams (video, voice and chat).
- \* All students have access to CodeBench (starting from lab 2).
- \* Instruction videos on using Teams and CodeBench will be made available in Wattle.

# Important TODOs

- \* Complete the **demographic information questionnaire**.
- \* **Sign up to a lab group.**
  - If there is no place free in any lab at any time that you can attend:
    - > don't sign up to a group you cannot attend;
    - > email `comp1730@anu.edu.au` with your ANU ID, a complete list of all groups that you can attend, and any preference.
  - Labs only start in semester week 2.
  - In-lab assessment starts in semester week 4.



\* To activate your account on the CSIT computers, you must log into STREAMS:

1. `https://cs.anu.edu.au/streams/;`
2. log in with your ANU user id and password;
3. log out again.

Do this **at least 24 hours** before your first lab.