

COMP1730/COMP6730

Programming for Scientists

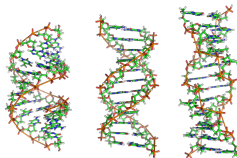
Introduction to the course
and administrative matters

Lecture outline

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

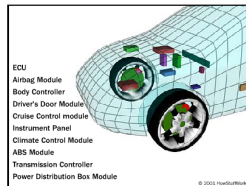
Why learn programming?

- * 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.



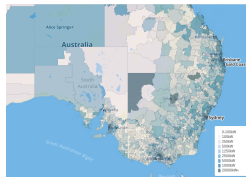
- ★ Technical systems increasingly run on software.

- A modern car has over 30 computers, running >10,000 lines of code.



- ★ Simulation and optimisation are needed for large-scale design questions.

- Intermittent renewables account for around 7.5% of energy generation. How do we design the power grid to work with 50%?



- * As 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 is the 1st (IEEE) – 4th (Tiobe) most popular language,
- * particularly for science and engineering uses.
- * Open source, available on most platforms.
- * Many packages:
 - over 200 in the python standard library;
 - over 60,000 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.

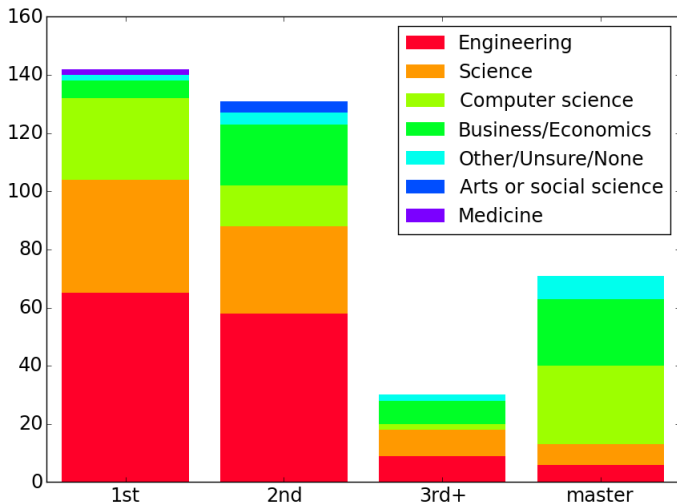
Learning outcomes

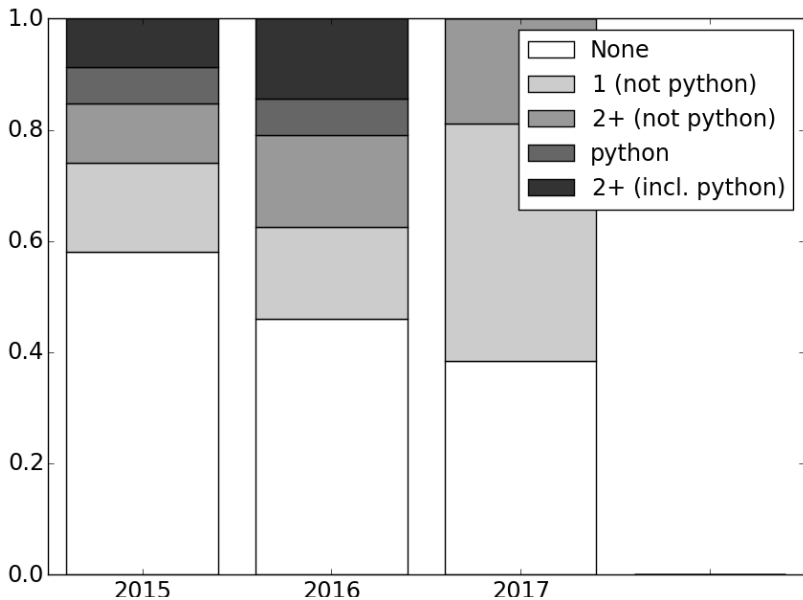
(revised from ANU Programs & Courses)

Students who succeed in all aspects of this course will:

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

About you: students in the course







Alternative courses

- * COMP1040 (“The Craft of Computing”)
 - Covers as much python, but at a much quicker pace.
 - Focus on tool use and libraries for data processing, analysis and visualisation.
 - You cannot take both 1040 and 1730!
- * COMP7230
 - Masters-only, data science-focused intro to programming using python
 - Delivered in online/intensive mode.

Course info & contacts

- * cs.anu.edu.au/courses/comp1730/
- * Wattle for forums, quizzes, surveys, assignment submission.
- * To ask a question:
 - Use the discussion forum on wattle.
 - For *personal* questions, use the course email: comp1730@anu.edu.au.
 - Tutors' contact info will be on wattle.
 - No office contact hours (ANU has not provided me with an office).

Discussion forum – 3 simple rules

1. **Read before you post.**

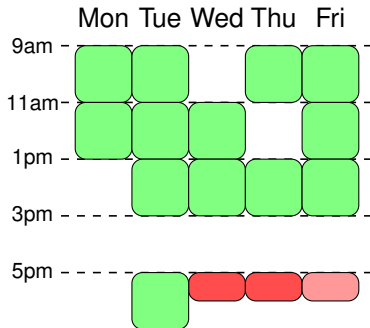
Before posting a question, check if your question has already been answered.

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 (or problem description).

Schedule overview



- * 2–3 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.
- * Complaints about the schedule?

dvc.academic@anu.edu.au

Assessment scheme

- * 5 small homework assignments (20%)
- * 1 larger project assignment (20%)
- * Final programming exam (15%)
- * Mid-sem. written exam (20%)
- * Final written exam (25%)

S. Week	
3	Homework 1 due (Monday) In lab: Questions on Hw 1
4	Homework 2 due (Monday) In lab: Questions on Hw 2
5	Homework 3 due (Monday) In lab: Questions on Hw 3
6	Homework 4 due (Monday) In lab: Questions on Hw 4
	Break
7	Homework 5 due (Monday) In lab: Questions on Hw 5 Mid-sem. written exam (TBC)
8	Project release
11	Project due
Exam period	Final written exam Programming exam

- * The complete assessment scheme is on the course web site at `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/special-assessment-consideration` **regarding special consideration.**

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 3.

* 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.

Student course representatives

- * Course representatives:
 - point of contact for fellow students who have issues/comments that they are not comfortable to raise with convenor directly;
 - participate in the SRC meetings a few times per semester.
 - Reps are encouraged to provide collective feedback directly to the convenor/lecturer.
- * Interested? Send your name and ANU id to `studentadmin.cecs@anu.edu.au` (don't forget to mention which course!)