

COMP1730/COMP6730

Programming for Scientists

Introduction to the course
and administrative matters



Announcements

- * **Read announcements made in the news forum on wattle**

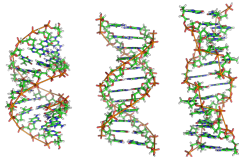


Lecture outline

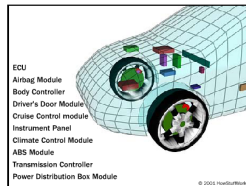
- * Why learn programming?
- * Course overview.
- * Info, contacts & schedule.
- * Assessment scheme.
- * Important TODOs.

Why learn programming?

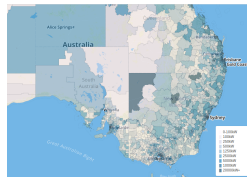
- * Science rests on data... *more and more data.*
 - The Australian SKA Pathfinder radio telescope outputs 2.5GB/s (the SKA is expected to be around 100 times more).
 - A human genome (around 3 billion base pairs) can be sequenced in 3 days.
- * Processing this data needs software.



- ★ Technical systems increasingly run on software.
 - A modern car has over 30 computers, running >10,000 lines of code.



- ★ Simulation and optimisation are needed to solve large-scale design challenges.
 - Intermittent renewables produced ~8.25% of Australia's electricity in 2017. How do we design the 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 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_rate = interest_rate/12
    # interest rate is given in % so need to divide by 100
    r = monthly_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.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.

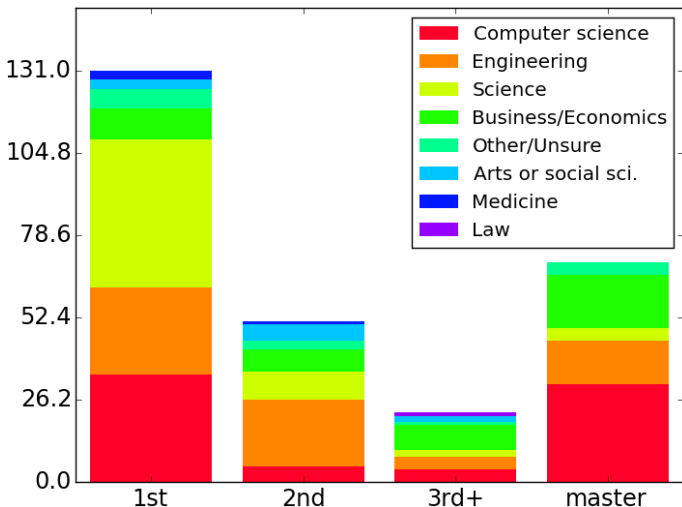
Learning outcomes

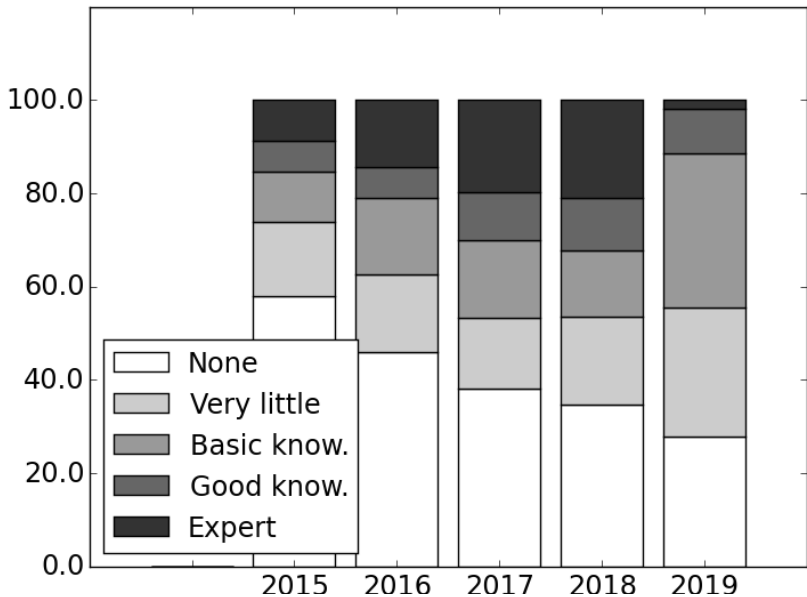
(revised from ANU Programs & Courses)

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.

About you: students in the course





Course info & contacts

- * 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`.
 - *Always use your ANU email.*
 - Tutors' contact info will be on wattle.

Discussion forum – 3 simple rules

1. **Read before you post.**

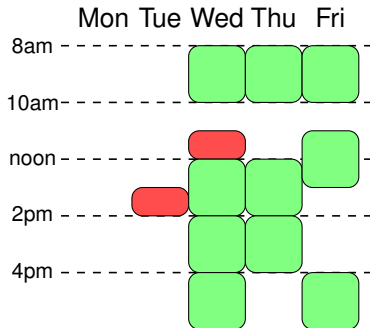
Before posting a question, check if your question has already been answered.

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 the 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 (preliminary)

- * 5 small homework assignments (15%)
- * 1 larger project assignment (25%)
- * Mid-semester exam (20%)
- * Final exam in 1 or 2 parts (40%)

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

- * 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/ regarding deferred assessments and special consideration.

Academic honesty

- * Homeworks are *individual*. You must write your own code, and understand every aspect of what you have written.
- * The project assignment may be done in small groups.
 - Collaboration (including copying solutions) between groups is *not* permitted.
 - The assignment will also have an individual component, which you must do by yourself.

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, and a complete list of all labs that you can attend.
 - 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? Write to comp1730@anu.edu.au or talk to me after the lecture.