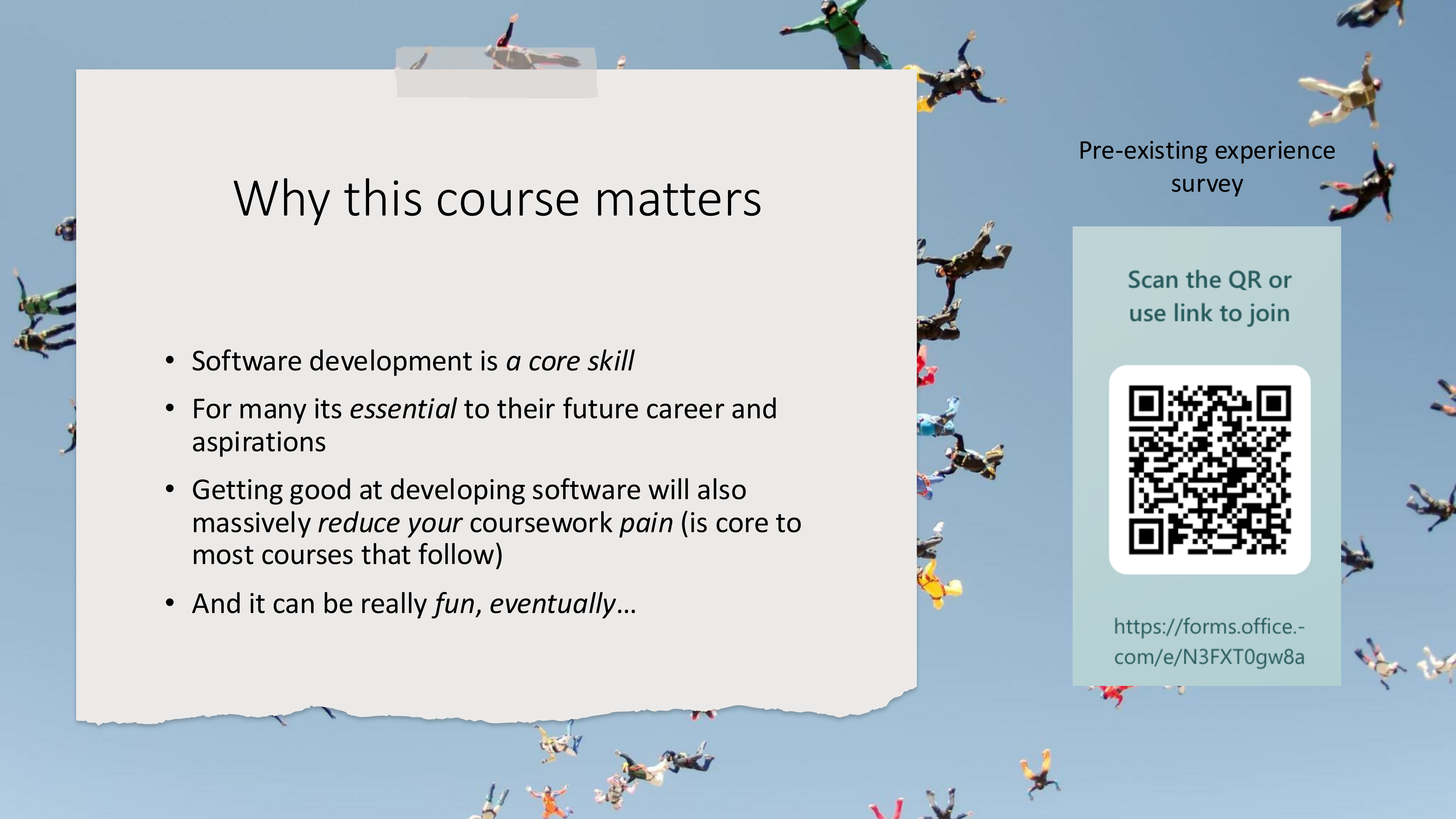
SCC.111 Software Development

Adrian Friday, Nigel Davies, Hansi Hettiarachchi, Saad Ezzini



Why this course matters

- Software development is *a core skill*
- For many its *essential* to their future career and aspirations
- Getting good at developing software will also massively *reduce your* coursework *pain* (is core to most courses that follow)
- And it can be really *fun*, *eventually*…

Pre-existing experience survey

Scan the QR or use link to join

https://forms.office.com/e/N3FXT0gw8a

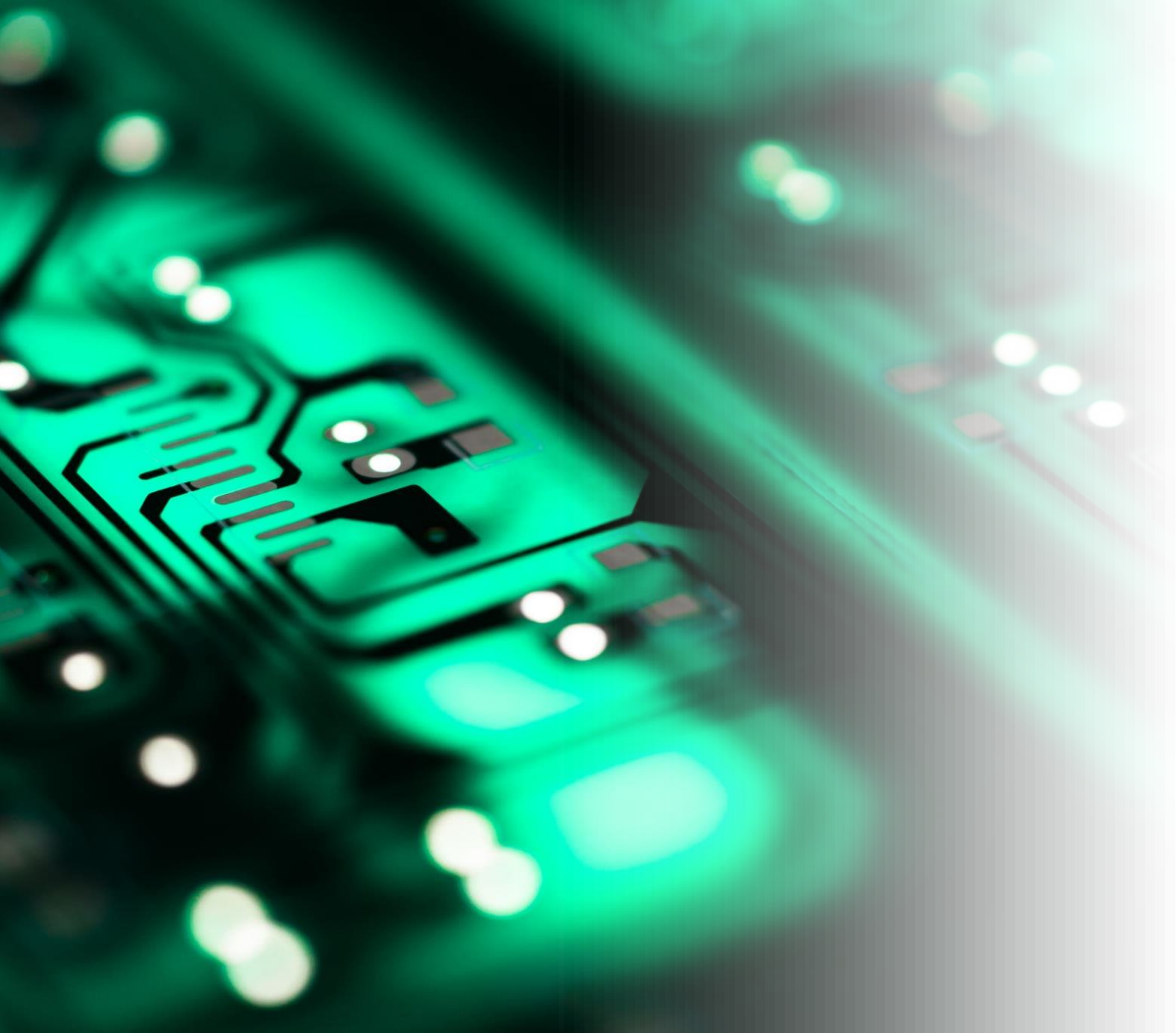
More than ‘just a programming course’

“This module aims to instil the knowledge, understanding and skills expected of a principled computer programmer. More specifically it aims to develop a coherent understanding of the **principles and practice** of **imperative programming**, the software development lifecycle and its associated **tools** and **techniques**.”

https://tinyurl.com/scc111spec

programming is the tool you use to realise your and other people's dreams !

programming … it’s like playing the guitar



programming is fiercely creative and collaborative – you are creating new things that didn’t previously exist!

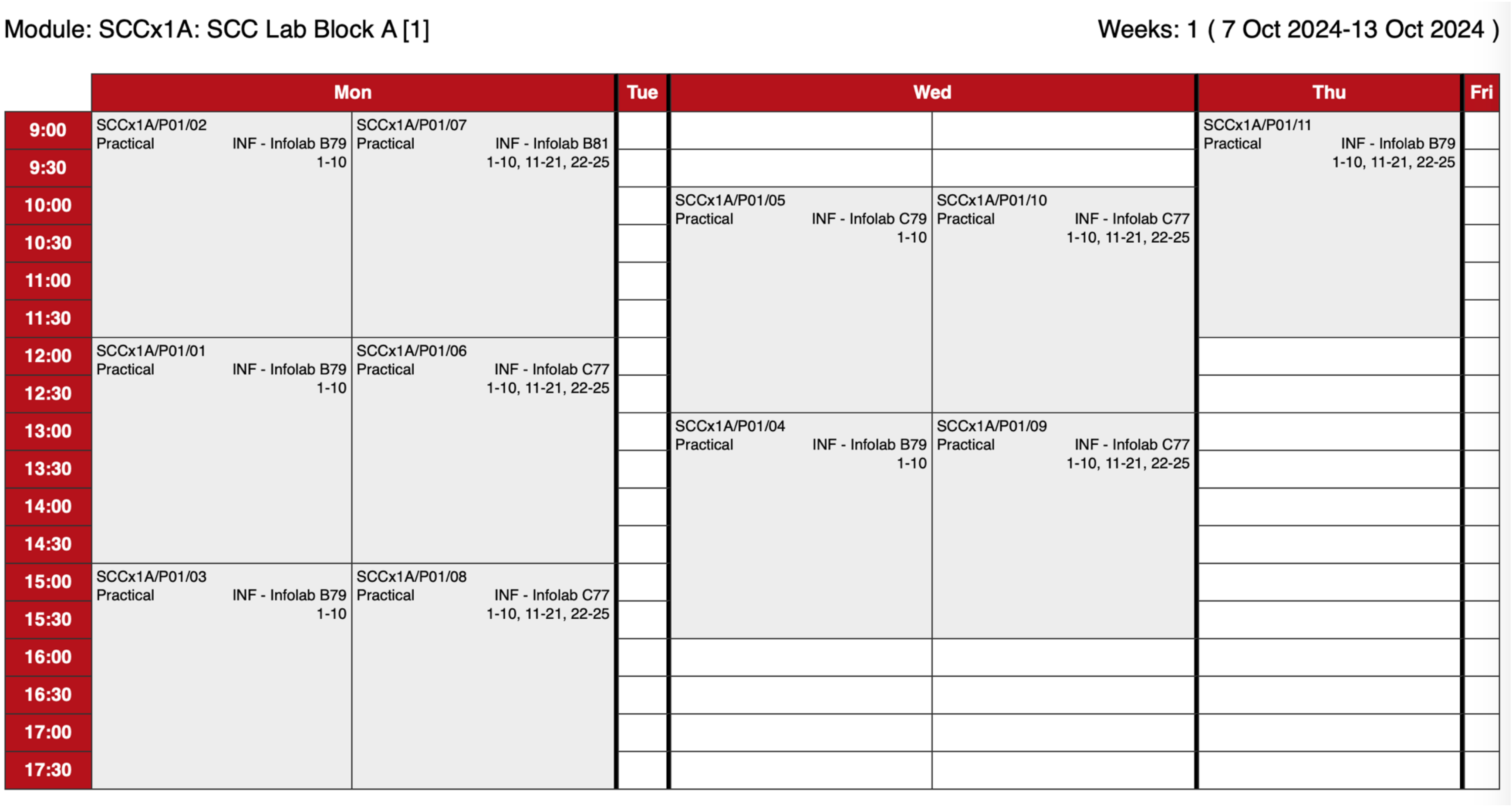
Who is teaching you?

- Hansi Hettiarachchi, Adrian Friday, Saad Ezini, Nigel Davies – probably about 120+ years of combined programming experience… ☺

How this Course is Taught

- There are **two** lectures per week (repeated once)
- You ***must*** attend your practical class each week
- You **may** attend other practicals if there is space and you need more help
- You **should** expect to work outside the labs and seek help using the moodle forum (please respect working hours) – the lab will not be enough for most tasks…

Course timetable (majors)

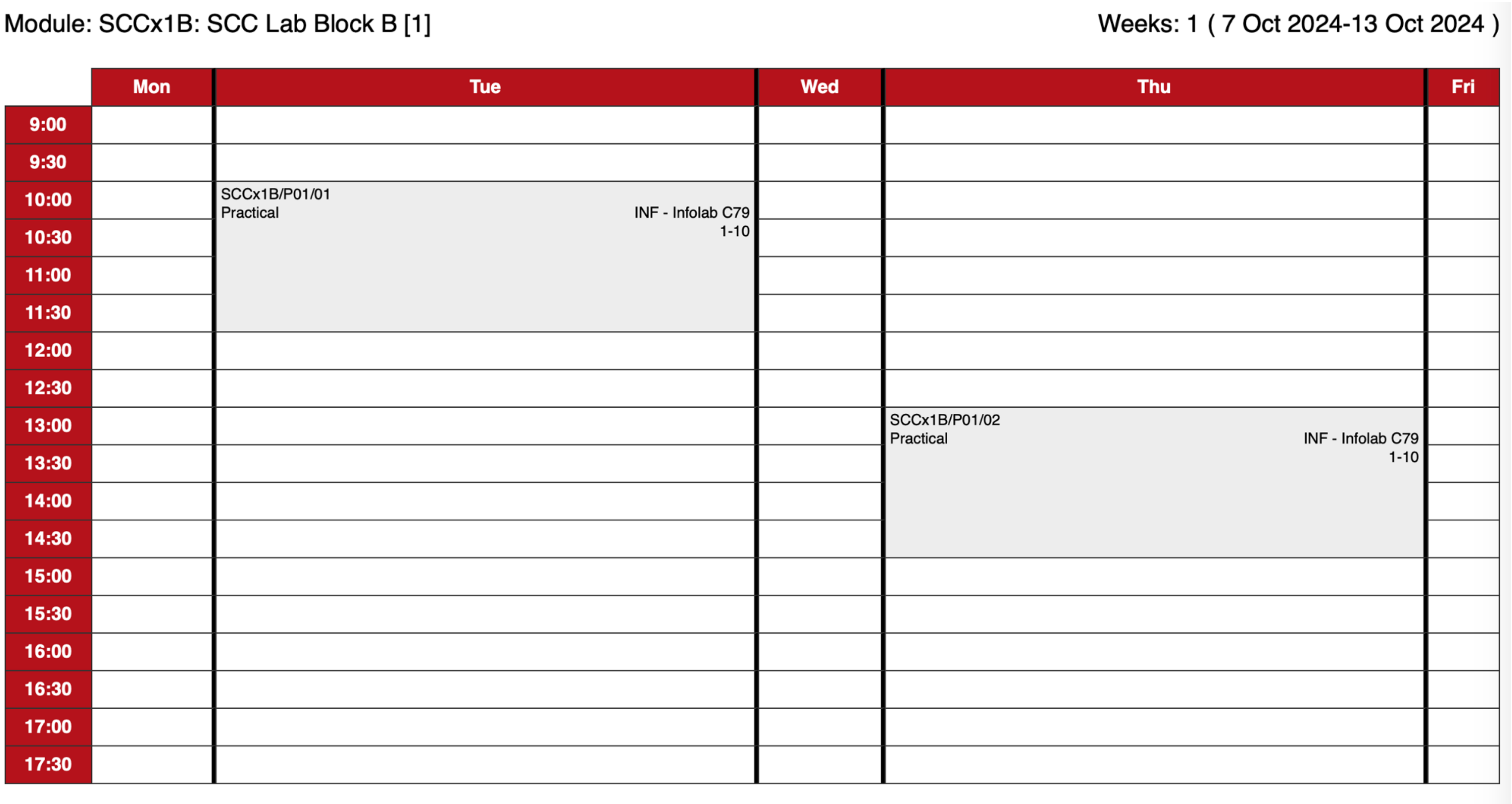


Module: SCCx1A: SCC Lab Block A [1]

Weeks: 1 (7 Oct 2024-13 Oct 2024)

	Mon		Tue	Wed		Thu	Fri
9:00	SCCx1A/P01/02 Practical INF - Infolab B79 1-10	SCCx1A/P01/07 Practical INF - Infolab B81 1-10, 11-21, 22-25				SCCx1A/P01/11 Practical INF - Infolab B79 1-10, 11-21, 22-25	
9:30							
10:00				SCCx1A/P01/05 Practical INF - Infolab C79 1-10	SCCx1A/P01/10 Practical INF - Infolab C77 1-10, 11-21, 22-25		
10:30							
11:00							
11:30							
12:00	SCCx1A/P01/01 Practical INF - Infolab B79 1-10	SCCx1A/P01/06 Practical INF - Infolab C77 1-10, 11-21, 22-25					
12:30							
13:00				SCCx1A/P01/04 Practical INF - Infolab B79 1-10	SCCx1A/P01/09 Practical INF - Infolab C77 1-10, 11-21, 22-25		
13:30							
14:00							
14:30							
15:00	SCCx1A/P01/03 Practical INF - Infolab B79 1-10	SCCx1A/P01/08 Practical INF - Infolab C77 1-10, 11-21, 22-25					
15:30							
16:00							
16:30							
17:00							
17:30							

Course timetable (minors)



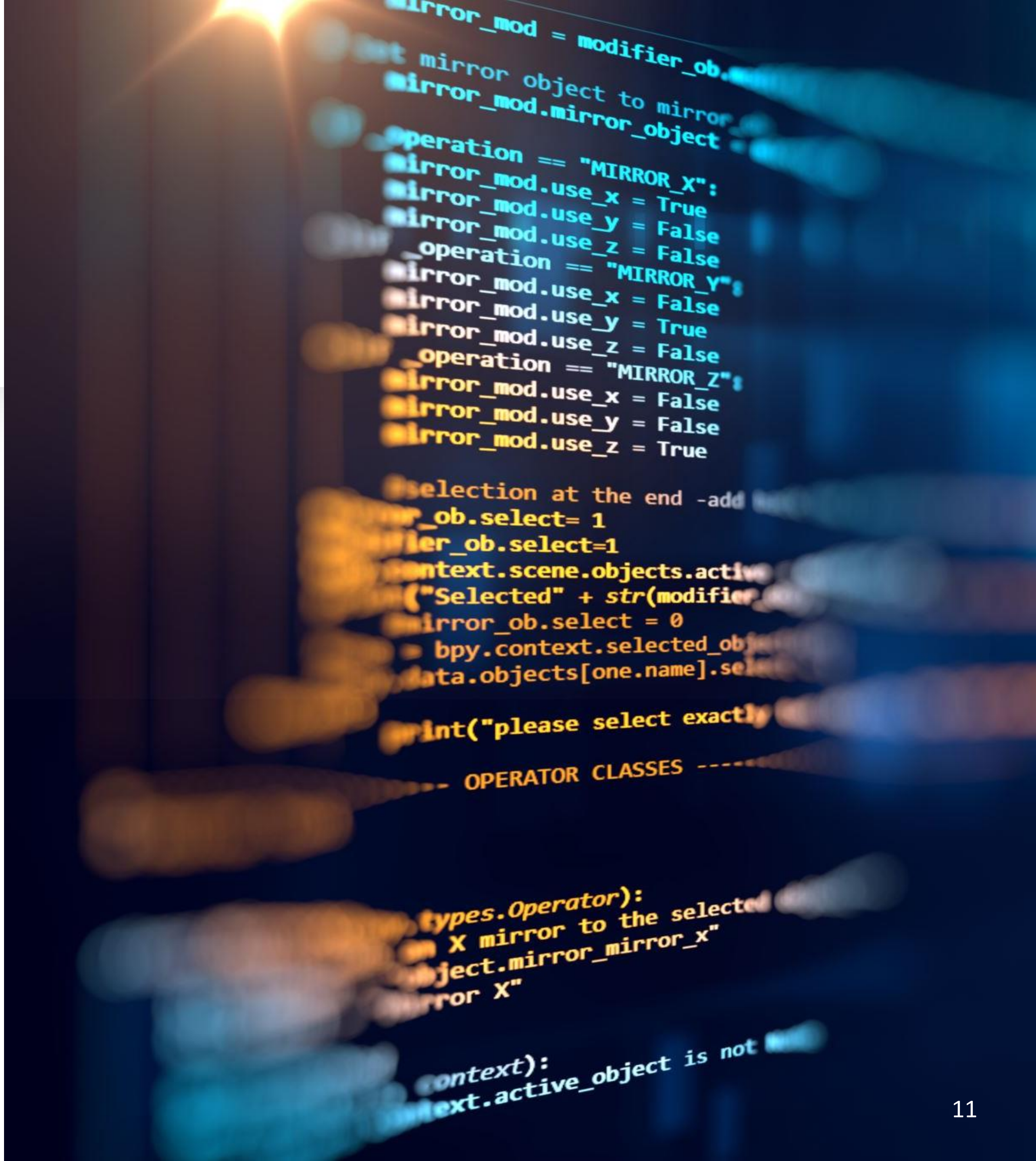
Module: SCCx1B: SCC Lab Block B [1]

Weeks: 1 (7 Oct 2024-13 Oct 2024)

	Mon	Tue	Wed	Thu	Fri
9:00					
9:30					
10:00		SCCx1B/P01/01 Practical INF - Infolab C79 1-10			
10:30					
11:00					
11:30					
12:00					
12:30					
13:00				SCCx1B/P01/02 Practical INF - Infolab C79 1-10	
13:30					
14:00					
14:30					
15:00					
15:30					
16:00					
16:30					
17:00					
17:30					

How this Course is Assessed

- There is an exam (70%) and coursework (30%) – the coursework helps you **pass!**
- Coursework consists of:
  - on-line tests and programming activities (week 5, 10, 15, 20)
  - a more open ended coding project (weeks 21-25)
- Exam is during the summer term and **is worth a lot**
- Coursework is submitted online and checked for plagiarism automatically. *We catch multiple cases each year*



```
irror_mod = modifier_ob.
 mirror object to mirror
irror_mod.mirror_object
peration == "MIRROR_X":
irror_mod.use_x = True
irror_mod.use_y = False
irror_mod.use_z = False
_operation == "MIRROR_Y":
irror_mod.use_x = False
irror_mod.use_y = True
irror_mod.use_z = False
_operation == "MIRROR_Z":
irror_mod.use_x = False
irror_mod.use_y = False
irror_mod.use_z = True

election at the end -add
_ob.select= 1
er_ob.select=1
ntext.scene.objects.active
("Selected" + str(modifier
irror_ob.select = 0
 bpy.context.selected_obj
ata.objects[one.name].sel

int("please select exactly

- OPERATOR CLASSES ----

types.Operator):
 X mirror to the selected
ject.mirror_mirror_x"
rror X"

context):
ext.active_object is not
```

Learning culture

A mix of experience and ability in the class (from 0 to lots!)

If you’ve little or no experience, that’s **ok** – try *not to panic* and don’t worry if others are ahead, that’s normal!

If you’ve more/ lots, do the more advanced exercises, help others

A word on academic integrity

- The course is designed to help you learn
  - It needs to be ‘your work’, not LLM/AI, not your friends’, not someone on the Internet, your family, your partner, or your hyperintelligent dog
  - Working on non-group work or individual assessments as a group *is malpractice*
  - By all means discuss, learn and study from each other, but no code or answers should be exchanged!
  - *All submitted code is checked for plagiarism! Integrity matters!*

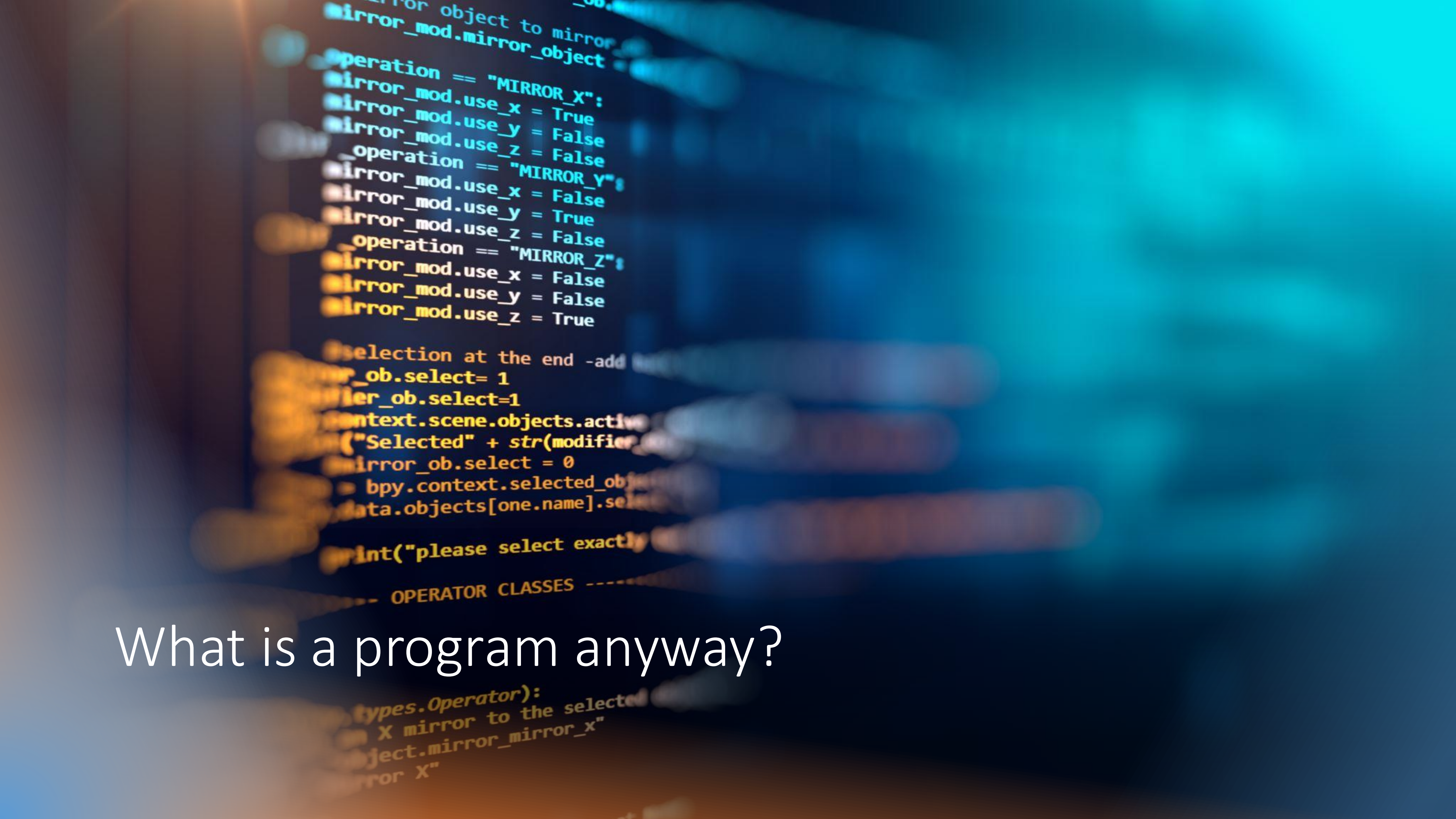
Learning through feedback and ‘good learning practice’

- **Formative** lab exercises – show us your code in each weekly lab session for detailed feedback
    - **keep (paper) notes** so you can reflect and improve week on week
- **Summative** Assessment (Quizzes, Project)
    - Quiz marks back normally the following week (barring handling of extensions etc.)
    - Project coursework due week 25, marks back within 4 weeks
- If you have extenuating circumstances, **talk to the SCC Teaching Office (TO)**

We want to see what **you** can do

- Integrity (your work, no faking results) and aspire to do well (effort!):
  - Keep up to date with the course (well, all courses ;))
  - Attend
  - Get the textbook (second hand/ https://onesearch.lancaster-university.uk/) – going beyond will only benefit you
  - Check understanding by asking us questions – *there are no silly questions*
  - Start the coursework **when it is set**, not **when it is due**

What is a program anyway?

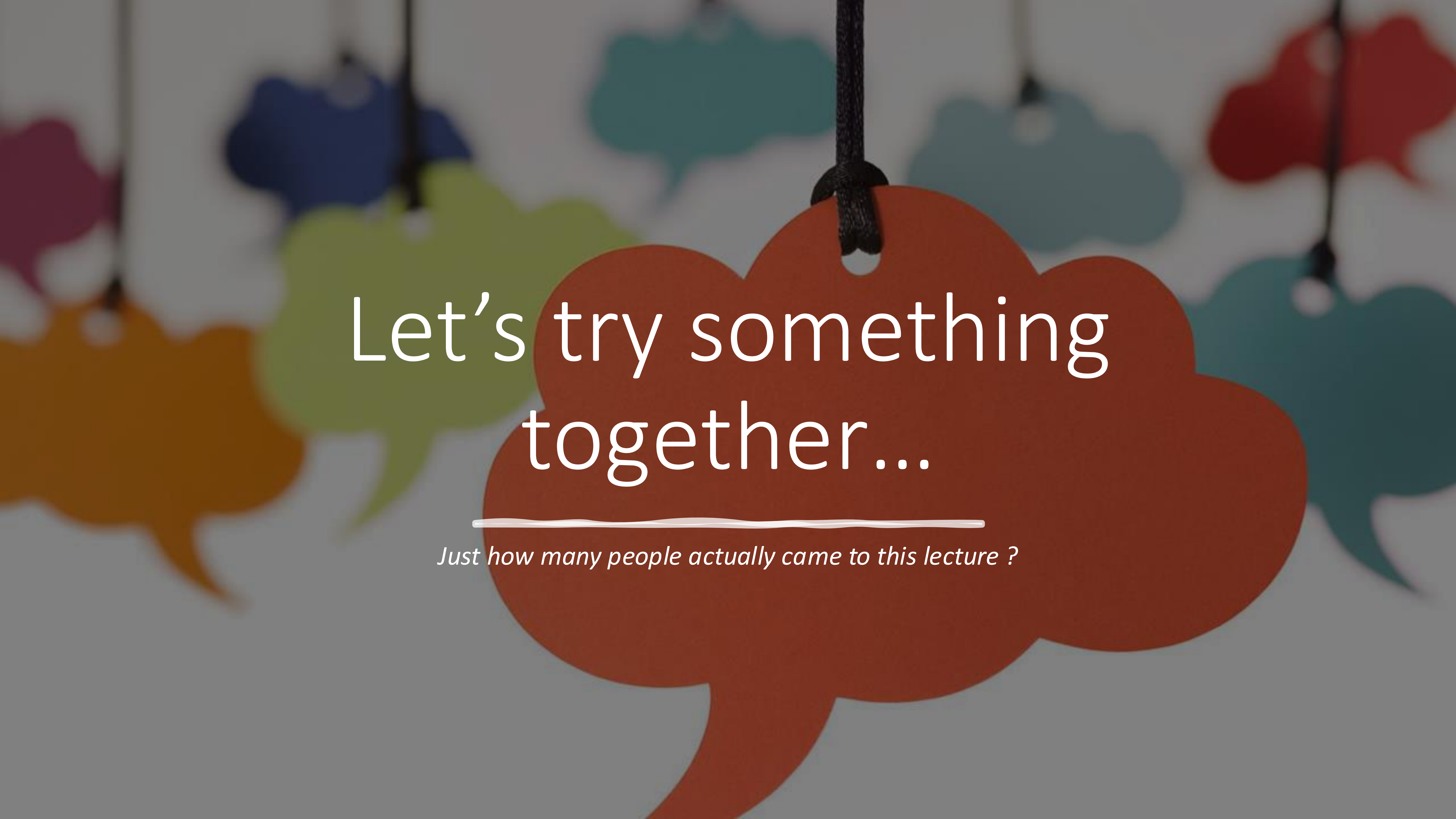


```
rror object to mirror_ob
mirror_mod.mirror_object
operation == "MIRROR_X":
mirror_mod.use_x = True
mirror_mod.use_y = False
mirror_mod.use_z = False
_operation == "MIRROR_Y":
mirror_mod.use_x = False
mirror_mod.use_y = True
mirror_mod.use_z = False
_operation == "MIRROR_Z":
mirror_mod.use_x = False
mirror_mod.use_y = False
mirror_mod.use_z = True

selection at the end -add
_ob.select= 1
er_ob.select=1
ntext.scene.objects.active
("Selected" + str(modifier
irror_ob.select = 0
= bpy.context.selected_ob
ata.objects[one.name].se

int("please select exactly

-- OPERATOR CLASSES ----

types.Operator):
X mirror to the selected
ject.mirror_mirror_x"
ror X"
```

Consider this…

- **STEP 1:** Heat oven to 190C/fan 170C/gas 5. Butter two 20cm sandwich tins and line with non-stick baking paper.
- **STEP 2:** In a large bowl, beat 200g caster sugar, 200g softened butter, 4 beaten eggs, 200g self-raising flour, 1 tsp baking powder and 2 tbsp milk together until you have a smooth, soft batter.
- **STEP 3:** Divide the mixture between the tins, smooth the surface with a spatula or the back of a spoon.
- **STEP 4:** Bake for about 20 mins until golden and the cake springs back when pressed.

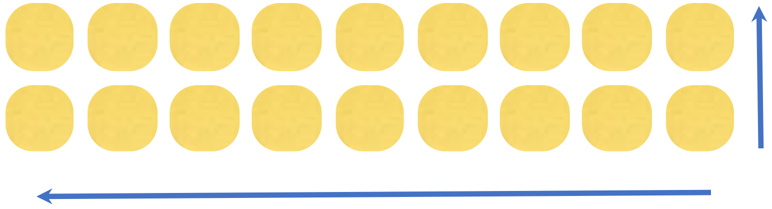
Recipe courtesy: https://www.bbcgoodfood.com/recipes/classic-victoria-sandwich-recipe



Let’s try something together…

Just how many people actually came to this lecture ?

How many people are at this lecture ?



linear time complexity

The Algorithm

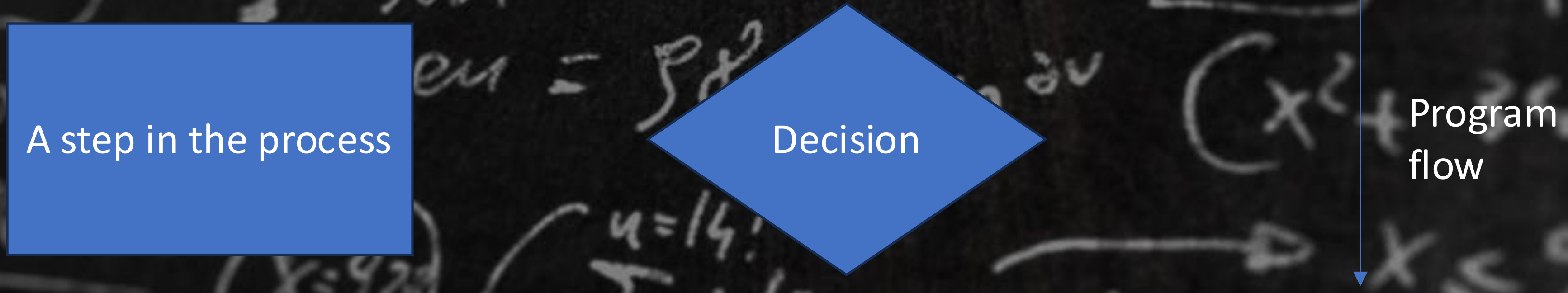
- Set counter to zero
- While not at last seat, do…
- If someone in the seat, add one to counter
- Move onto next seat
- End▫

- What features does our algorithm illustrate?

Scan the QR or use link to join

https://forms.office.-
com/e/4cHjz6ndrx

BTW - that was nearly a program :) It was certainly an algorithm…



A step in the process

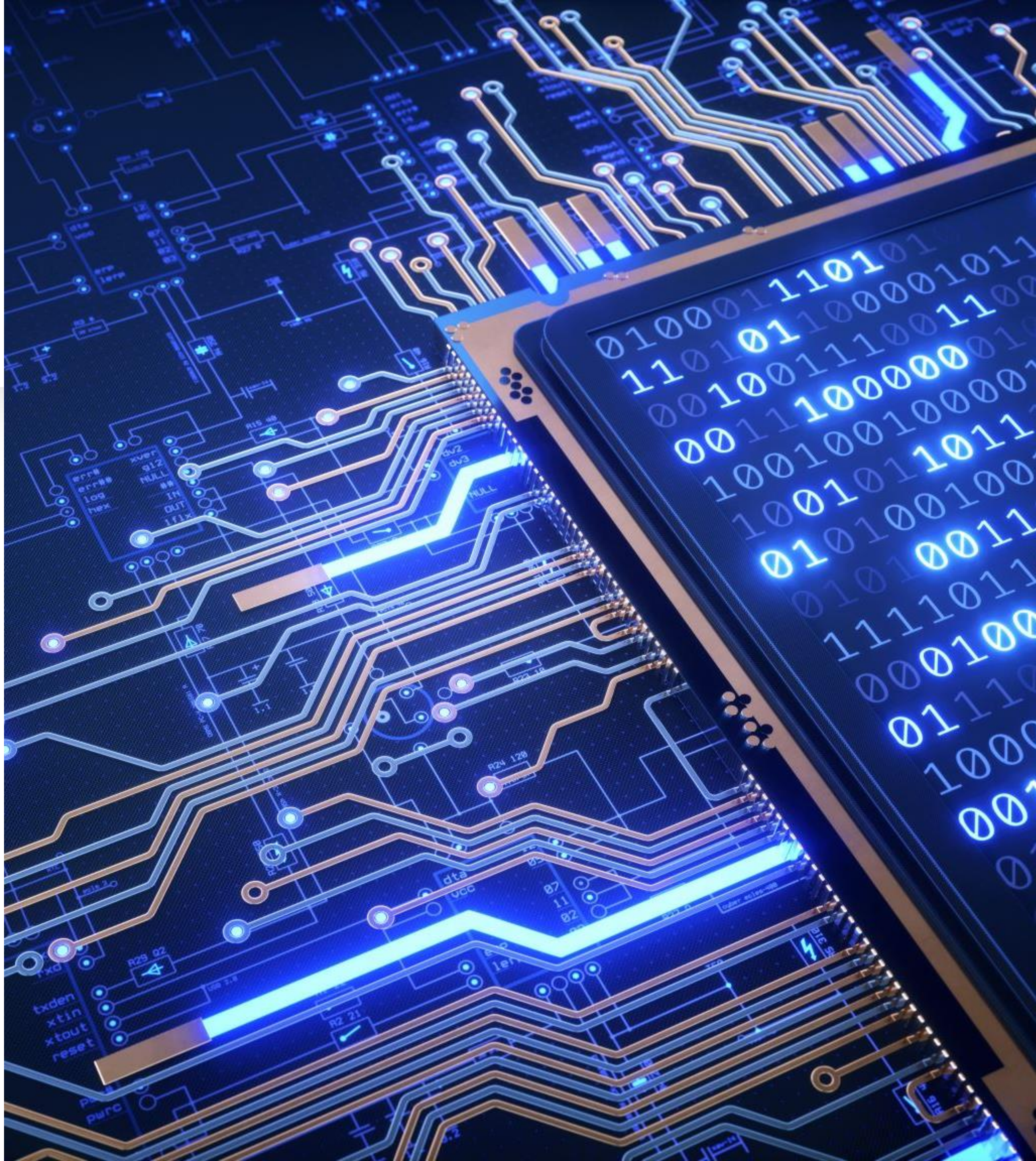
Decision

Program flow

A program…

“A detailed plan or procedure for solving a problem with a computer”

more specifically, “an unambiguous, ordered sequence of computational instructions necessary to achieve such a solution.” *courtesy, Encylopedia Brittanica.*

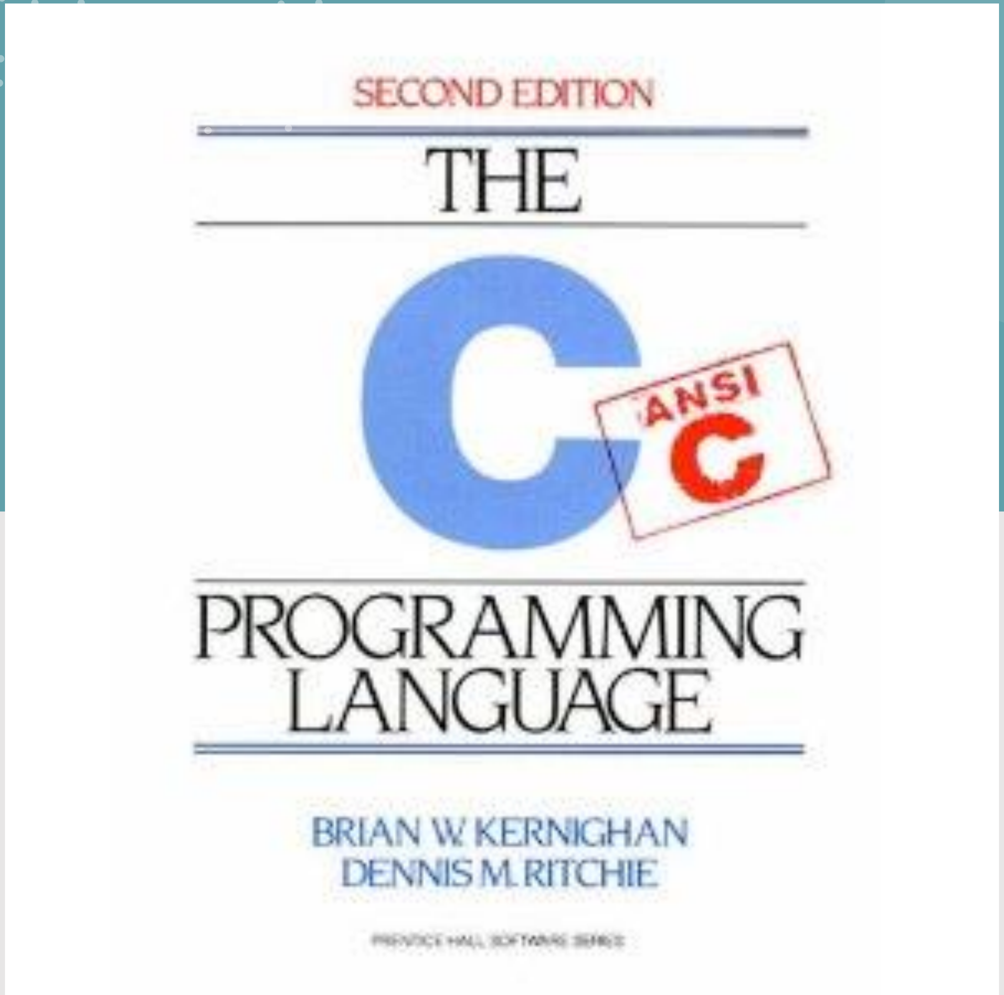


Imperative programming…

- In computer science, **imperative programming** is a programming paradigm of software that uses statements that change a program's state (Wikipedia)
  - So, a critical step is to think through what to represent (**what**)
    - – a total, a brightness, a geo-location, a time, an audio sample, a command for a robot…
  - As well as, how our program should manipulate this (**how**)
  - We're going to practice this ***a lot***

“C”

This term we start with an imperative programming language called **C**. First created in 1970s to build UNIX. C is compact, low-level, and is used to generate fast, efficient code that exploits hardware features well.



SECOND EDITION

THE C PROGRAMMING LANGUAGE

ANSI C

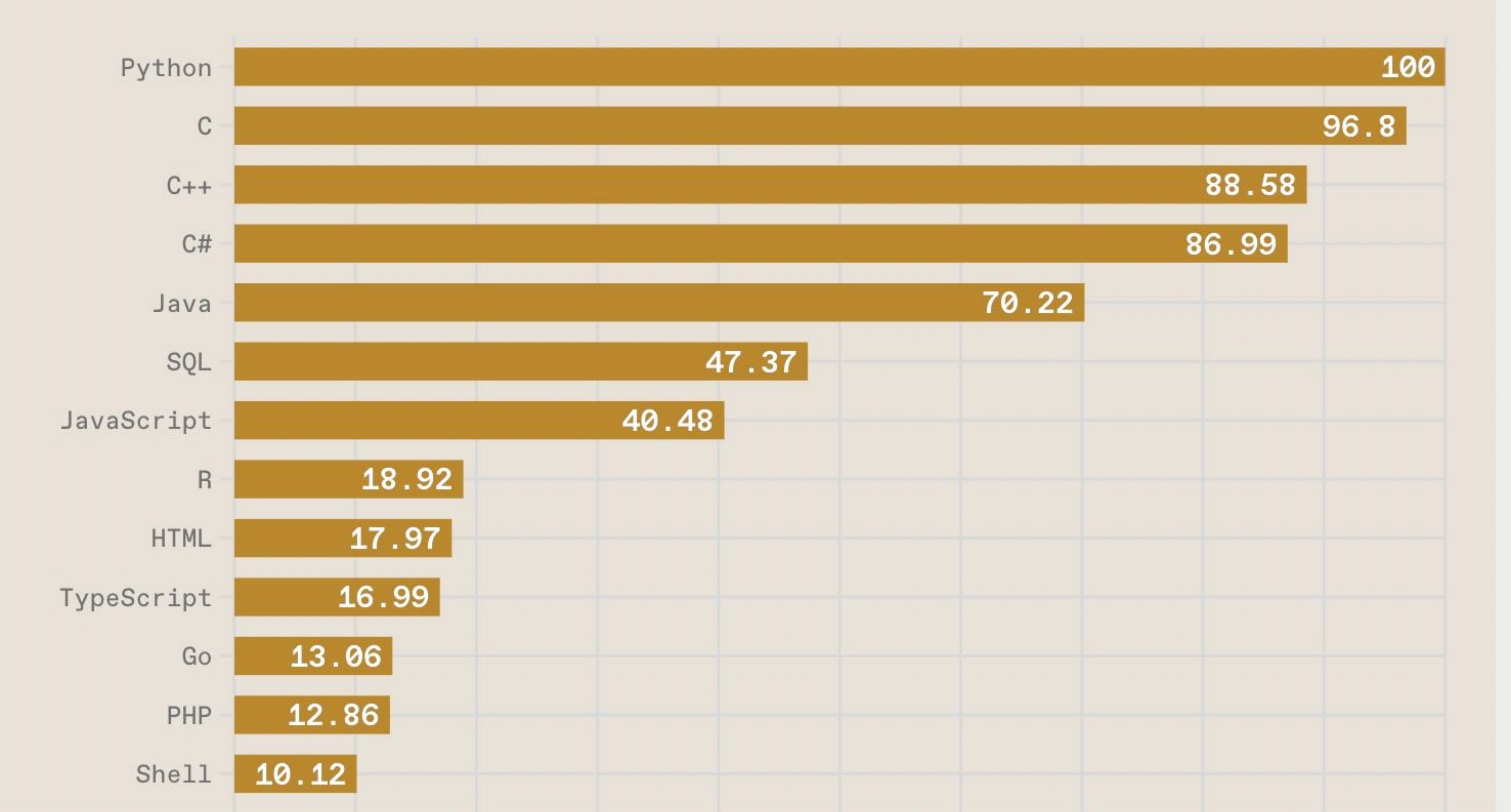
BRIAN W. KERNIGHAN
DENNIS M. RITCHIE

PRENTICE HALL SOFTWARE SERIES

C is the ancestor, the parent, the inspiration – *very much still with us!*

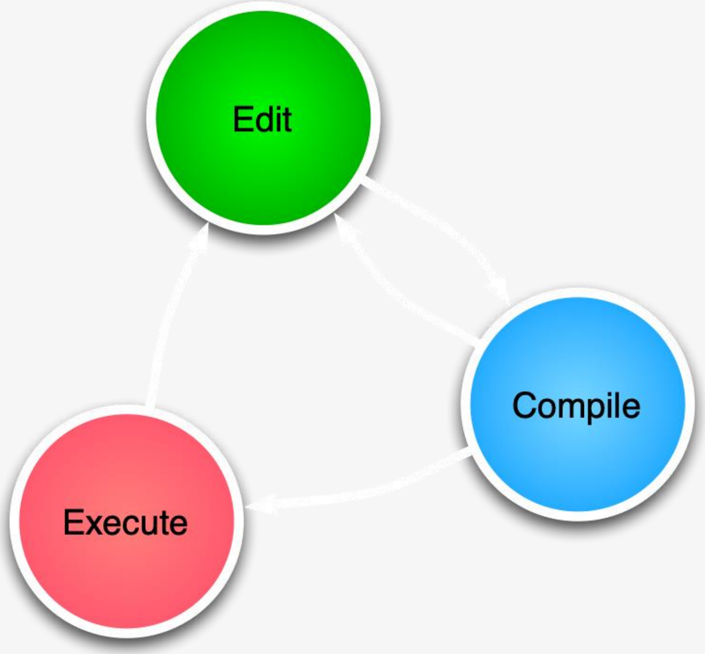
- Windows 10 (2015) – 55 million lines of code (loc)
- Windows 11 60-100m estd
  Linux - ~15m
  Mac OS X ~85m

  The kernel is written in what language?


Python
100
C
96.8
C++
88.58
C#
86.99
Java
70.22
SQL
47.37
JavaScript
40.48
R
18.92
HTML
17.97
TypeScript
16.99
Go
13.06
PHP
12.86
Shell
10.12

C is a ‘compiled language’

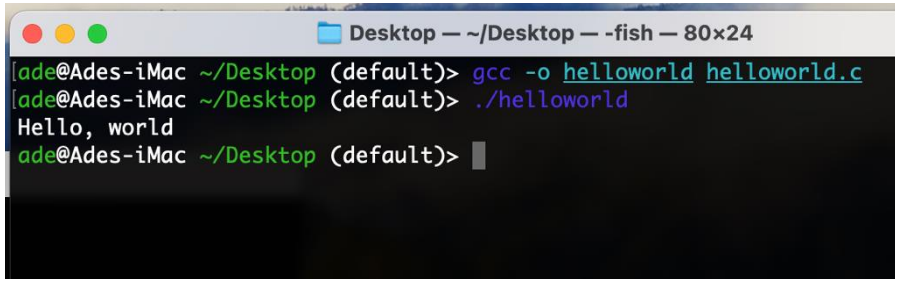
- In C we will use standard tools to form a “tool chain”, we
  - use **a text editor** to write and edit the code
  - a **compiler** which translates the code into something the computer can understand (only if the program syntax is correct!)
  - a resulting **executable** we can run
- In C there is an *explicit* compilation phase where syntax is validated and low-level executable code is created *iff* syntax is correct



Meeting our first C program

```
#include <stdio.h>

int main()
{
  printf("Hello, world\n");
}
```

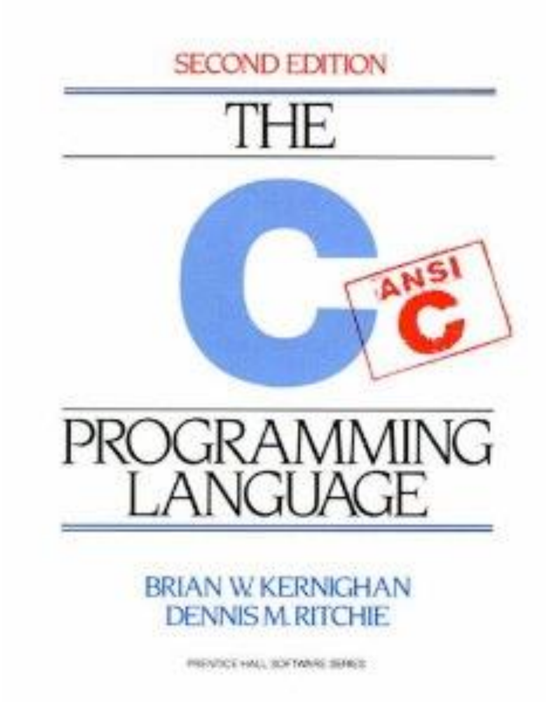
```
$ gcc -o helloworld helloworld.c
```



```
[ade@Ades-iMac ~/Desktop (default)> gcc -o helloworld helloworld.c
[ade@Ades-iMac ~/Desktop (default)> ./helloworld
Hello, world
ade@Ades-iMac ~/Desktop (default)> 
```

The C Programming Language (2nd Edition)

Brian W. Kernighan and Dennis Ritchie



The **definitive** Language Reference

Available online via the library collection, see course moodle page and, https://onesearch.lancaster-university.uk

Summary

- You should know how the course is structured, how it is assessed and how you’ll get feedback
- You should really, really want to be a great programmer :)
- You should be really looking forward to your lab sessions and getting started with C!
- Next lecture: *the key building blocks of C*