

INF1-IP

Introduction to Programming

Lecture 0 – Welcome to the course!



> Who we are



Fiona McNeill
Course Organiser
and lecturer



Matteo Degiacomi
lecturer



Rayo Verweij
lecturer



**Dominik
Grabarczyk**
Course TA



**Mario Navarro
Veiga**
Course TA and
tutor

> My Dog



His name is Dembi

> Your pet



Send me pictures of
your pets

Every week we will have
a series of pets of the
week

! What is this course for?

To lay the foundations of a good programming practice –
for:

- Students who have never coded
- Students who have done some coding in school
- Students who have maybe done tonnes of coding outside school but not using SE skills (what does that mean??)

It is intended to bring you up to *approximately* the level of A-level students in coding

It is about laying down an understanding of *how to programme* so you can develop your own practice.

! What is this course for?

- This is a practical course to help you create the basis of your programming practice
- The things you *learn* in this course are only useful to the extent that you are able to put them into *practice*
 - *Knowing things doesn't make you a good programmer; programming makes you a good programmer.*
- The most important thing to do in this course is to practice!

> But ... AI can do this for me, can't it?

Yes, AI could definitely ace this course.

So why are we making you do it?

If you only ever want to do very simple coding, you don't really need to learn it → get AI to do it

But if you want to develop code at a higher level, the future is in *partnering* with AI.

That means you need to *understand* what it is doing.

- You need to understand code at a deep and intuitive level
- To do this, you need to spend **time** building your expertise from the bottom up

> A non-coding example

We used AI to produce the original materials for this course
... a whole course produced in an hour – hurrah!

But the materials weren't great.

To get to the standard we were happy with took *a lot* of work

Was it easier than creating the course from scratch?

Probably, but we were only able to produce a good course from the AI-generated materials *because we know what this looks like and how to do it.*

> Good code v functioning code

- Writing code that appears to achieve the results you want is not the same as writing *good* code
- Good code must be:
 - Easy to read and understand - for you (down the line) and for others
 - Easy to debug
 - Easy for somebody else to take over, use, add to, etc.
- Code that works but doesn't do these things isn't much use
- There's a lot of bad code out there! We don't need more bad code in the world.

> Technical debt

This is accumulated when you write code *fast* but not *well*

- + you can get results really quickly
- the code is really hard to reuse, share and understand

Technical debt means you are borrowing time now -- developing quickly -
- but creating a time-sink down the line -- when you need to build on this code.

AI *massively* increases the amount of technical debt in the world

Agentic code builds massive technical debt unless well supervised by someone who understands code.

> Code hacking vs Software Engineering

Writing code that works is an important part of this course - but not the only part.

You also need to think about code design, documenting your code and communicating about code to coders and non-coders alike - otherwise code is not very useful.

You will be assessed on your ability to do this, not just on coding!

How this course works



> Course structure

- Lectures:
 1. Double slot on Monday afternoons 1600-1800
 - a) Student worksheets to work through **during** lectures (whether live or recorded) – this usually start fairly empty
 - b) Slides with key points on
 - c) Recordings of lectures (available afterwards)
 2. Full worksheets to help you learn on your own
 3. Extra online material to complement your learning and build your practice

These complement each other and reinforce your learning. The live lectures should be your **main** focus

> Course structure

- Tutorials:
 - Every week from Week 2, on Fridays
 - You should be with your academic family in your tutorial (occasionally this isn't possible)
- Drop-in lab support:
 - Available from InfPals

> Mastery Learning

- The course is a **mastery learning** course.
- This means that there are different sections of the course, and you can only go on to later parts when you have shown **mastery** of the earlier parts.
- Why?
 - This approach is recognised as a good way to teach Intro to Programming, because:
 - Students learn at different rates
 - In programming, mastering the basics is essential for understanding more advanced programming

> Mastery Learning

- There are **three parts** to the course
 - Part 1: Weeks 1-5
 - Part 2: Weeks 6-8
 - Part 3: Weeks 9-10
- Each part is assessed by a *class test* and an *assignment*
 - The class test shows you can *make things work*; the assignment shows you *understand the underlying principles*
 - For each, there are three grades:
 - **E- Excelling** – demonstrates mastery
 - **P - Pass** – demonstrates mastery
 - **N - Needs work** – does not demonstrate mastery

> Mastery Learning

- For each part, you demonstrate that you have mastered that part by:
 - Getting a P or an E on the class test, **and**
 - Getting a P or an E on the assignment
- If you get an N on either or both, you have not yet mastered all of that part
 - You will get feedback on which parts you still need to work on
 - You will retake whichever part you did not pass at the next assessment point.

> Mastery Learning

- There are **three parts** to the course
- There are **four** assessment points – so you can retake once and still get full grades. If you need to retake more than once, you are no longer able to get full grades but can still get a high score.
- You cannot take any assessment at Level 2 until you have passed **both** the class test and the assignment at Level 1 (and so on for Level 3).
 - E.g., in the first sitting you pass Class Test 1 but fail Assignment 1 → you revise part 1 rather than progressing to part 2
 - In the second sitting you do not do a class test (because you've already passed Class Test 1 but have not started part 2)
 - You retake Assignment 1.

	C1-N	C1-P	C1-E	C2-P	C2-E	C3-P	C3-E
A1-N	20	25	30	X	X	X	X
A1-P	25	40	45	50	55	X	X
A1-E	30	45	50	55	60	X	X
A2-P	X	50	55	60	65	70	75
A2-E	X	55	60	65	70	75	80
A3-P	X	X	X	70	75	80	90
A3-E	X	X	X	75	80	90	100

C = class test
A = assignment

Levels:
E = excelling
P = pass
N = needs work

> Example path

Lily has done some coding before she starts uni.

She works well across the semester and goes along to tutorials

She passes both Class Test 1 and Assignment 1 in the first sitting.

She then goes on to the second part and passes Class Test 2 and Assignment 2 at the next sitting.

She then does the final part and passes both Class Test 3 and Assignment 3 at the next sitting.

She will get 80-100% (depending how well she passed) and does not have to do the final assessment in revision week.

She does not do the final assessment in revision week

> Example path

Claude has done some coding before he starts uni but has a hard time finding accommodation and doesn't focus on his courses much to begin with.

He passes Class Test 1 but fails Assignment 1 in the first sitting → he has not shown mastery of the first part of the course.

He then revises the parts of the course he had not mastered over the next two weeks, and passes Assignment 1 at the next sitting (he does not need to do a class test that week).

He then continues to the second part and passes both Class Test 2 and Assignment 2 at the next sitting.

He wants a chance to get full marks, so takes both Class Test 3 and Assignment 3 in the extra session during revision week. If he passes, he will get 80-100%; if not, he will get 60-70%

> Example path

Matthew has never coded and struggles to understand the concepts initially. This is normal because people learn at different rates.

He fails Class Test 1 and Assignment 1 in the first sitting → he has not shown mastery of the first part of the course.

He then revises the parts of the course he had not mastered over the next two weeks, but still struggles and fails Class Test 1 again, although he passes Assignment 1.

He then continues to revise, and at the third sitting retakes Class Test 1 and passes.

Now that he has passed the course, he chooses not to do more assessment in revision week, but instead to focus on revising for his other courses. He will get 40-50% (depending on how well he has passed).

> Example path

Scarlett follows the same path as Matthew, but decides that she does want to attend assessment for the second part of the course in revision week. She pass the Class Test 2 but not Assignment 2, and will get 50-60%.

> Example path scheduleStudent

Student	Week1	Week2	Week3	Week4	Week5	Week6	Week7	Week8	Week9	Week10	Week 11	Week12
Lily	Part 1	Part 1	Part 1	Part 1	Part 1	Part 2	Part 2	Part 2	Part 3	Part 3		
Claude	Part 1	Part 1	Part 1	Part 1	Part 1	Part 1	Part 1	Part 2	Part 2	Part 2	Part 3	Part 3
Matthew	Part 1	Part 1	Part 1	Part 1	Part 1	Part 1	Part 1	Part 1	Part 1			
Scarlett	Part 1	Part 1	Part 1	Part 1	Part 1	Part 1	Part 1	Part 1	Part 1	Part 2	Part 2	

> What if you don't finish the course ..?

- You can always progress if you get 40% in a course
 - Usually, we don't know what that says about what you have learned or not learned
 - If you get 40% in IP we know exactly what you know well and what you haven't done
- Lecturers in later courses will know what all students **should definitely** know and will not have to make guesses / assumptions
- The later material is always available for you to use on your own, and for later-years lecturers to use to help develop your skills

> How does this work with lectures and tutorials?

- Lectures:
 - Live lectures will teach:
 - Part 1: Wks 1-5
 - Part 2: Wks 6-8
 - Part 3: Wks 9-11
 - If you are retaking sections, you should use the recordings and the complete worksheets to practice the elements you still have to master
- Tutorials:
 - You will be grouped in tutorials with other students at the same part of the course
- Drop-in InfPals sessions:
 - Always useful to attend, but especially important when you are revising
 - Will support you whatever part of the course you are currently at

> How do I know how I'm doing?

Whenever you go into our assessment system, STACK, it will tell you what you have passed so far

We will show you how to do this later on

You will not be able to access assessment for later parts until you have passed earlier parts, so you cannot get confused about what to do.

We will be talking about this a lot, so don't worry if it doesn't make sense right now!

! Summer resit

Unless you have been given permission to opt-out of IP, you **must** pass it to progress to second year

You will have lots of opportunity to retake the core parts of it, so we hope not many will fail

But if you do, you will be offered a resit in the summer. We don't know yet exactly how this will work, but, like the semester assessment, we will assess all three parts, and you will need to pass part 1 to pass, and other parts to get higher grades.

> Where to find the material

- The OpenCourse page is the main source of Information
- You need to log in to Learn to access
 - Assessment
 - EdStem – the first time you use it
 - Notable – the first time you use it
- You can access EdStem and Notable via OpenCourse once you have created the correct credentials to do this by accessing via Learn