THE UNIVERSITY of EDINBURGH
informatics

http://tinyurl.com/inf1-handbook
https://informaticsstudentlearningsupport.github.io

Welcome to Year 1

Michel Steuwer - michel.steuwer@ed.ac.uk

THE UNIVERSITY of EDINBURGH
informatics

http://tinyurl.com/inf1-handbook
https://informaticsstudentlearningsupport.github.io

Michel Steuwer - michel.steuwer@ed.ac.uk

Under **G**raduate Year 1 Organiser: coordinate between UG1 courses

My teaching:

Compiling Techniques - Year 3 course

My research:

Programming Languages and Compilers

THE UNIVERSITY of EDINBURGH
informatics

http://tinyurl.com/inf1-handbook
https://informaticsstudentlearningsupport.github.io

You: the 2021 UG1 cohort

- > 350 students
- > 100 students from outside the UK
- > 100 countries represented in the school

THE UNIVERSITY of EDINBURGH
informatics

http://tinyurl.com/inf1-handbook
https://informaticsstudentlearningsupport.github.io

UG Degrees in the School of Informatics

- Computer Science (and …)
- Artificial Intelligence (and …)
- Software Engineering
- Informatics
- Cognitive Science

Degree	Numbers
CS (BSc)	123
CS (BEng)	63
CS + math	46
CS + mgm.	6
CS + physics	3
AI + CS	52
AI	12
SE	19
MInf	16
Cog Sci	14

http://tinyurl.com/inf1-handbook
https://informaticsstudentlearningsupport.github.io

Program for this talk

- **Robotics in Edinburgh** (Chris) Xiaoxuan Lu
- **First Year Courses** Michel Steuwer
- **Programming Club** Michio Honda
- **Informatics Connect** Fiona McNeill

THE UNIVERSITY of EDINBURGH
informatics

http://tinyurl.com/inf1-handbook
https://informaticsstudentlearningsupport.github.io

First Year Courses

Informatics Courses *(all degrees)*

- **INF1A: Introduction to Computation**,
  20 Credits, Semester 1
- **INF1B: Object Oriented Programming**,
  20 Credits, Semester 2

Maths Courses *(almost all degrees)*

- **Introduction to linear algebra**,
  20 Credits, Semester 1
- **Calculus and its applications**,
  20 Credits, Semester 2

Cognitive Science Courses *(CogSci degree)*

- **INF1-CG: Cognitive Science**
  20 Credits, Semester 2

Courses outside the School of Informatics (40 credits)
- Chosen from anywhere in the University (timetable permitting)

THE UNIVERSITY of EDINBURGH
informatics

http://tinyurl.com/inf1-handbook
https://informaticsstudentlearningsupport.github.io

Course selection via PATH: https://path.is.ed.ac.uk

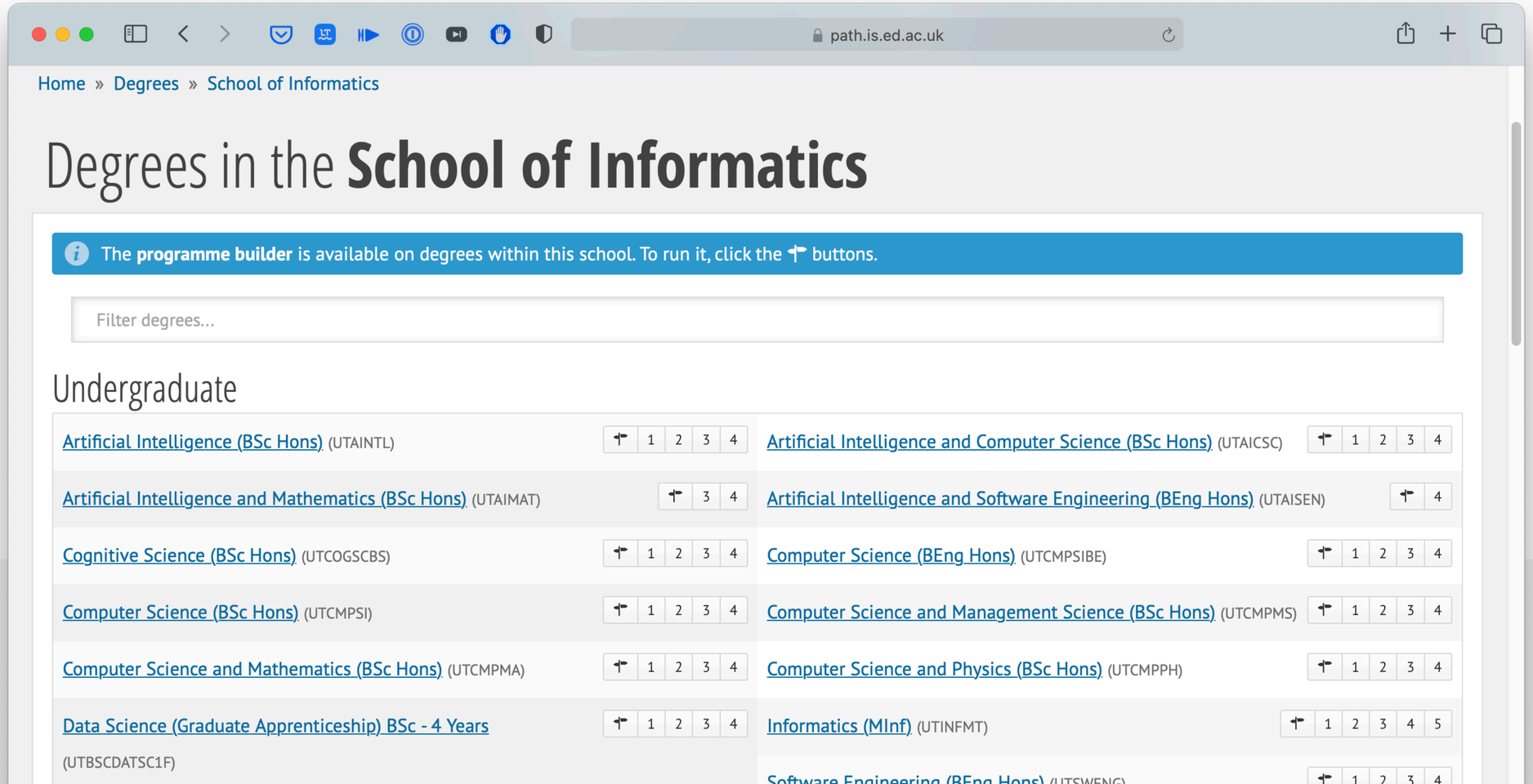
Discuss your selection with your Personal Tutor

`goto` https://path.is.ed.ac.uk;

`select` **“DEGREES”** at the top;

`select` **“School of Informatics”**;

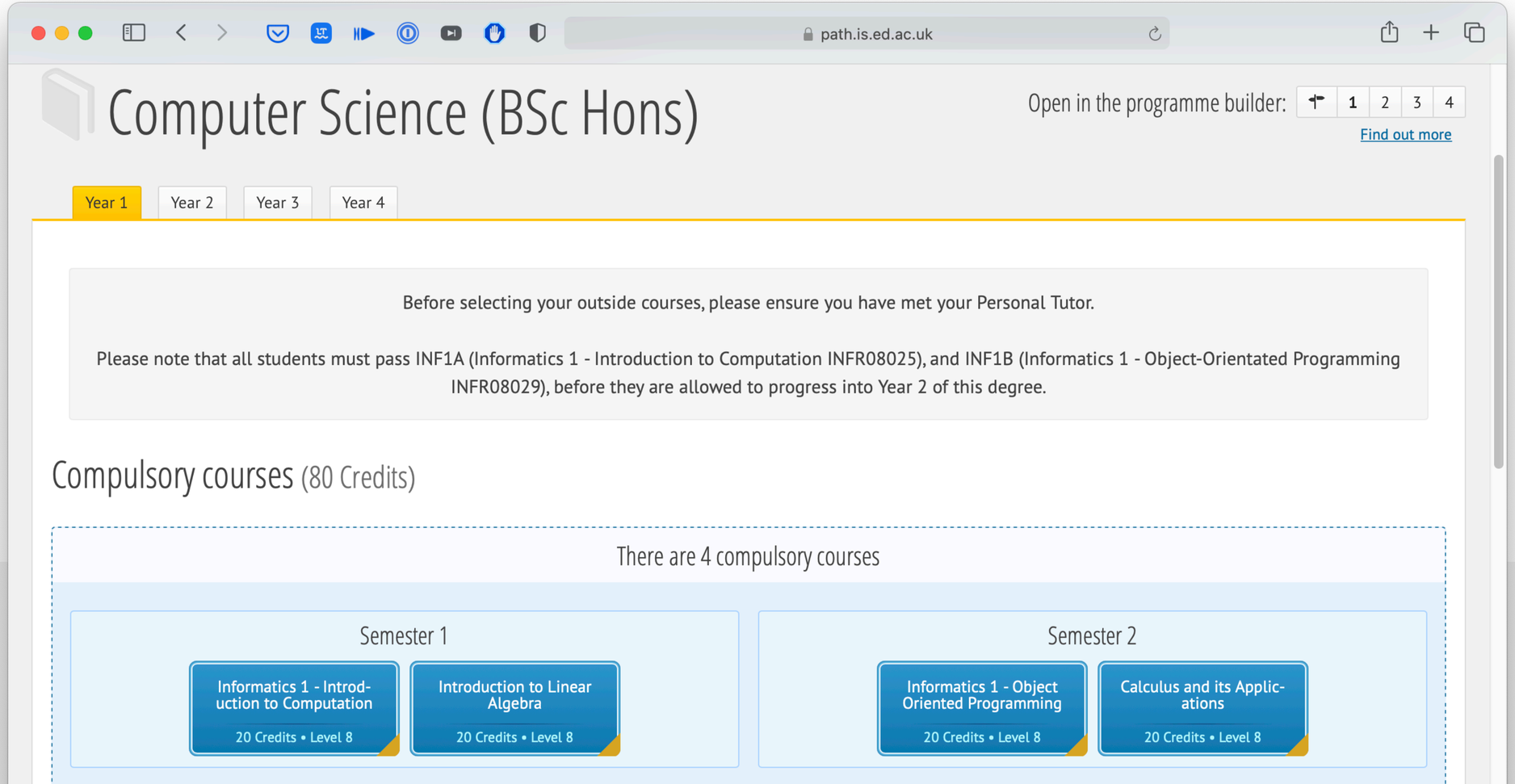
`select` your degree program —>



THE UNIVERSITY of EDINBURGH
informatics

http://tinyurl.com/inf1-handbook
https://informaticsstudentlearningsupport.github.io

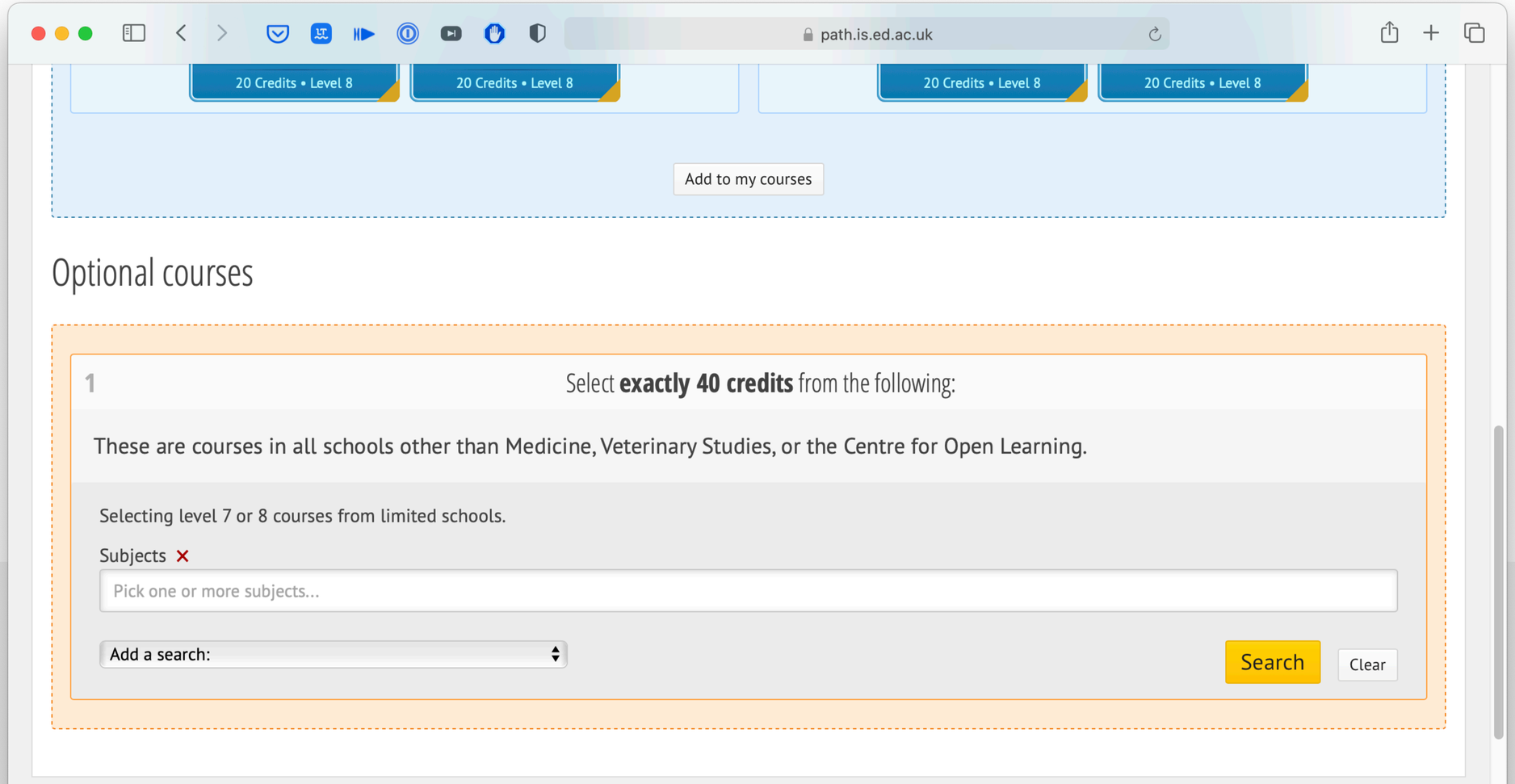
Course selection via PATH: https://path.is.ed.ac.uk



THE UNIVERSITY of EDINBURGH
informatics

http://tinyurl.com/inf1-handbook
https://informaticsstudentlearningsupport.github.io

Course selection via PATH: https://path.is.ed.ac.uk



THE UNIVERSITY of EDINBURGH
informatics

http://tinyurl.com/inf1-handbook
https://informaticsstudentlearningsupport.github.io

INF1A: Introduction to Computation

Consists of two parts: Computation and Logic & Functional Programming

Computation and Logic studies two fundamental topics:

- ***Finite State Machines*** provide a simple model of computation that is widely used. We will study their structure and behaviour.
- ***Propositional Logic*** is the first step in understanding logic. We use it to build computational tools for solving puzzles and reasoning about informatics systems and their properties.

We study the basics of **functional programming** using the language **Haskell**

- (Pure) functional programs have no state and no "side-effects"
  - The output from a function depends only on the input
  - This makes it much easier to predict the programs behaviour
- The course introduces fundamental concepts such as recursion, abstraction, higher-order functions and data types.
- And emphasises the practical use of these constructs …

Julian Bradfield

Philip Wadler

THE UNIVERSITY of EDINBURGH
informatics

http://tinyurl.com/inf1-handbook
https://informaticsstudentlearningsupport.github.io

INF1B: Introduction to Object Oriented Programming

General principles of programming in imperative and object-oriented languages

- learn to program in a major real-world programming language: **Java**
  - Many libraries and existing applications are written in Java
- Uses an ***imperative, object-oriented paradigm*** in contrast to Haskell
  - "Object-oriented" languages encapsulate that data, together with the code which manipulates them into "objects"
- This course tries to be very flexible to cater for very wide range of prior experiences, from no previous experience (or failing Inf1a …) to lots
  - To make that work, it's essential to go in ready to be in charge of your own learning and pick appropriate challenges

Fiona McNeill

http://tinyurl.com/inf1-handbook
https://informaticsstudentlearningsupport.github.io

INF1-CG: Cognitive Science

Matthias Hennig

A lightweight, but broad introduction into the vast field of Cognitive Science for first year students

The course covers the following topics from a computational point of view:

- ***Language***
- ***Vision***
- ***Memory and Attention***
- ***Cognitive and Neuroscience***
- ***Reasoning and Generalisation***

The course is suitable both for informatics students (the content from cognitive psychology, linguistics and neuroscience will be new) and students from other fields of cognitive science such as psychology (the modelling and programming will be new, we will gentry introduce this).

Maths Courses in Year 1

Introduction to Linear Algebra *Semester 1*

This course picks up on ideas some of which are likely to be familiar from previous study: vectors, matrices and simultaneous equations, but carries them much further.

Calculus and its Applications *Semester 2*

This course presents an introduction to the two main branches of calculus: differential calculus and integral calculus. At the heart of both lies the notion of the limit of a function, sequence, or series.

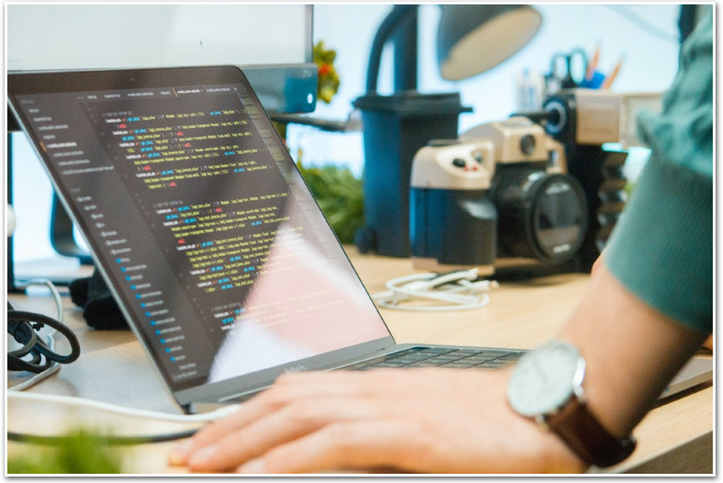
http://tinyurl.com/inf1-handbook
https://informaticsstudentlearningsupport.github.io

> *"**Programming** - and in general software development - is **not a purely theoretical skill, nor is it something you can do well without learning some fundamental concepts**. Unfortunately, far too often, teaching fails to maintain a balance between theory/principles and practicalities/techniques.*
>
> *Consequently, we see people who basically despise programming ("mere coding") and think that software can be developed from first principles without any practical skills.*
>
> *Conversely, we see people who are convinced that "good code" is everything and can be achieved with little more than a quick look at an online manual and a lot of cutting and pasting; ...*
>
> *My opinion is that both attitudes are far too extreme and lead to poorly structured, inefficient, and unmaintainable messes even when they do manage to produce minimally functioning code."*

Bjarne Stroustrup
designer of the
C++ programming language

http://tinyurl.com/inf1-handbook
https://informaticsstudentlearningsupport.github.io

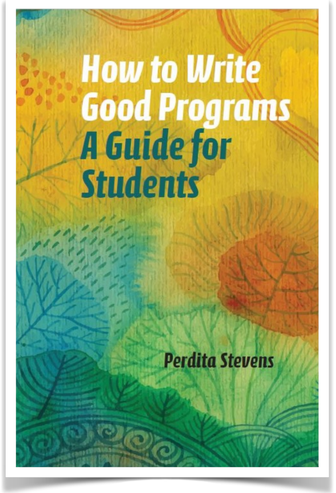
Programming is a fundamental skill



- *All Infomatics students* are expected to be able to program well
- Programming is not just “code hacking”
  - “real” programs need to be *correct*, *reliable*, *maintainable* & *efficient*
  - they are often large & complex, developing involves many people
  - this requires a solid understanding of the underlying principles and theory
- Programming is a practical craft
  - you can’t learn by just attending lectures & reading books!
  - you need to practice steadily outside of the formal course activities
- Different students will have vastly different previous experience
  - you may need to spend more, or less times on this than other students
  - don’t worry if other students seem to know more about his. They may not!
    Just work steadily on the course materials and take advantage of the support that we provide

http://tinyurl.com/inf1-handbook
https://informaticsstudentlearningsupport.github.io

> *People sometimes talk as though students could be divided into programming sheep and non-programming goats:* ***as though programming ability were innate****. My experience over more than twenty-five years of teaching, and most current research, suggest that this is* ***simply not true****. I have lost count of the number of times* ***I have seen students really struggle to begin with, perhaps even failing their first programming course, but go on to become excellent programmers****. […]*
>
> ***Some people love programming form the very beginning****. These people may have started coding at a young age, and often choose to sit up late into the night doing so. That's great […]. But, full disclosure:* ***I was not one of those people****. Indeed, when, as a child, I was first introduced to programming,* ***I didn't really see the point.*** *I didn't start to spend a lot of time on programming until, in my twenties, I encouraged a problem I couldn't solve without writing a program.* ***I learned to program because I had a problem I needed to solve, which I couldn't solve any other way.***"



How to Write Good Programs
A Guide for Students

Perdita Stevens
Prof. of Mathematics of Software Engineering

Class Representatives

Class representatives are appointed to represent the view of students

- By conveying student opinion about the courses, or by alerting staff to student issues that they are unaware of, class reps can make a real different to how Informatics 1 functions
  - gather feedback from students on all aspects of courses and facilities
  - alert staff to issues
  - Eligible to be selected as a member of Board of Studies or Teaching Committee
  - Attend weekly lunchtime meetings during semester
- **Volunteer as a class rep!** This is also a great opportunity for your CV
  - see **http://tinyurl.com/inf1-reps** for details
  - I will main around asking for volunteers; you will need to provide a short text
  - all students will be able to vote, but we will take diversity into account
  - please mail me if you have any questions

http://tinyurl.com/inf1-handbook
https://informaticsstudentlearningsupport.github.io

My Recommendations for your first year

- **Take responsibility for your own learning**
  - This may be different from the model at school that you are used to
- **All of the actives are designed to help you**
  - Take advantage of them
  - Prepare in advance
  - Participate; ask questions; make the activities work for you
- **Don't give up if it gets hard**
  - Some things are just hard & it takes a while — try again
- **If you are having difficulties, don't wait until it is too late!**
  - Discuss issues with someone (Lecturers & Personal Tutor) as soon as possible

Have fun!