Introduction
Jane Hillston, Head of School
Björn Franke, Director of Teaching

Robotics
Barbara Webb, Honours Project Co-ordinator

First Year Courses
Paul Anderson, UG1 Year Organiser

Programming Club
Michio Honda

Informatics Connect
Fiona McNeill
1st Year Course Content

Informatics Courses (40 credits)

- INF1a
  - Functional Programming
  - Computation and Logic  Semester 1
- INF1b Object-oriented Programming  Semester 2

Maths Courses (40 credits)

- Introduction to linear algebra  Semester 1
- Calculus and its applications  Semester 2

Outside Courses (40 credits)

- Chosen from anywhere in the University (timetable permitting)
  - See the handbook for details - look for the course web page
Communication

- Our natural languages are often verbose and ambiguous. This course introduces some tools that we use to talk and reason, clearly and unambiguously, about computational systems. We will study two fundamental topics:

Finite State Systems

- Finite state machines provide a simple model of computation that is widely used. We will study their structure and behaviour. They are used as basic computational models across the whole of Informatics and at the same time underpin many widely used applications and components.

Propositional Logic

- 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.

Michael Fourman
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 behaviour

- The course introduces fundamental concepts such as recursion, abstraction, higher-order functions and data types
- And emphasises the practical use of these constructs …
- Functional languages are becoming more popular in commercial applications
  - Barclays, Deutsche Bank, Facebook, Google, Intel (Haskell)
  - Netflix, Twitter, Foursquare, LinkedIn (Scala)

Philip Wadler
Object-Oriented Programming

General principles of programming in imperative and object-oriented frameworks using the language Java

- 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 the 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
A series of essential topics for Inf1A and ILA with screencasts, notes and associated online reading and quizzes within a book format embedded in Inf1A. The topics are arranged to help with relevant weeks in both courses but can be accessed at any time by student's choice.

For example, you may already have met number bases and modular arithmetic, but for many students this is a new topic. Or you may have done Set Theory in your home language, but English can be confusing with its use of 'and' /'or'. This will save you searching the internet for support material and supplement your learning.

Heather Yorston
“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)
All Informatics 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, involving 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 this. they may not! just work steadily on the course materials and take advantage of the support that we provide.
Class representatives are appointed to represent the views 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 difference 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 mail round for volunteers
    - you will need to provide a short blurb
    - all students will be able to vote
    - we will take diversity into account, as well as the vote results
    - please mail, or come and see me if you have any questions
Take responsibility for your own learning.
- This may be different from the model that you are used to

All of the activities 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 just are hard & it takes a while - try again

If you are having difficulties, don’t wait until it is too late
- Discuss it with someone as soon as possible

Have fun!
Introduction to informatics 1

Paul Anderson
dcs paul@ed.ac.uk

2020-2021

http://tinyurl.com/inf1-handbook