Lecture 1: Introduction

Dr John Levine

CS103 Machines, Languages and Computation
September 21st 2015

University of Strathclyde Science

The Rules of the Game

- CS103: Machines, Languages and Computation
- 20 credits, both semesters
- Two lecturers:
  - John Levine (Semester 1)
  - Anders Claesson (Semester 2)
- Assessed by 4 class tests (30%) and an exam (70%)
- Exemption from the exam is possible if you do well in the class tests

University of Strathclyde Science

The Rules of the Game

- Lectures are Mon 10 in RC 345, Fri 10 in RC 641
- Tutorials: Thu 10 in JA 505, Thu 1 in TG 227
- Lectures run Weeks 1-11 (no lectures in Week 12)
- Class materials will appear on the class web page:

  http://www.cis.strath.ac.uk/~johnl/CS103
- Facebook page: MLAC 2015
- Email: John.Levine@strath.ac.uk
- Room LT1420, extension 4524

University of Strathclyde Science

Weekly Exercises

- Every week, you do a classwork assignment consisting of reading and some pen-and-paper exercises.
- This should take about 1-3 hours and can be done on the train, in bed, between other lectures, etc.
- The classwork exercises are contained in a workbook which will be handed out in Lecture 2.
- By the end of Week 6 you should have completed all the exercises in the workbook.

University of Strathclyde Science

Class Tests

- The classwork is assessed by means of a class test: a mini-exam in the usual lecture slot
- There are 4 class tests for CS103: two this semester and two next semester
- The Semester 1 tests are on Friday 30th October and Friday 4th December.
- Feedback and results will be available 2 weeks after the test date.

University of Strathclyde Science

Exemption

- Each class test counts for 7.5% of your final mark for the class.
- If you pass all four tests and score an average over all four tests of 60% or more, then you are exempted from the degree exam.
- If you are exempt, your total mark for the class is the average of your marks in the four class tests.
- If you are not exempt, the degree exam counts for 70% of your mark and the 4 class tests for 30% of the mark.

University of Strathclyde Science

Course Aims

From the syllabus page:

To help students to achieve a broad knowledge of the essence of computation and computational systems, as embodied by the notions of computable functions, formal languages and recursion, logic and computability and abstract machines.

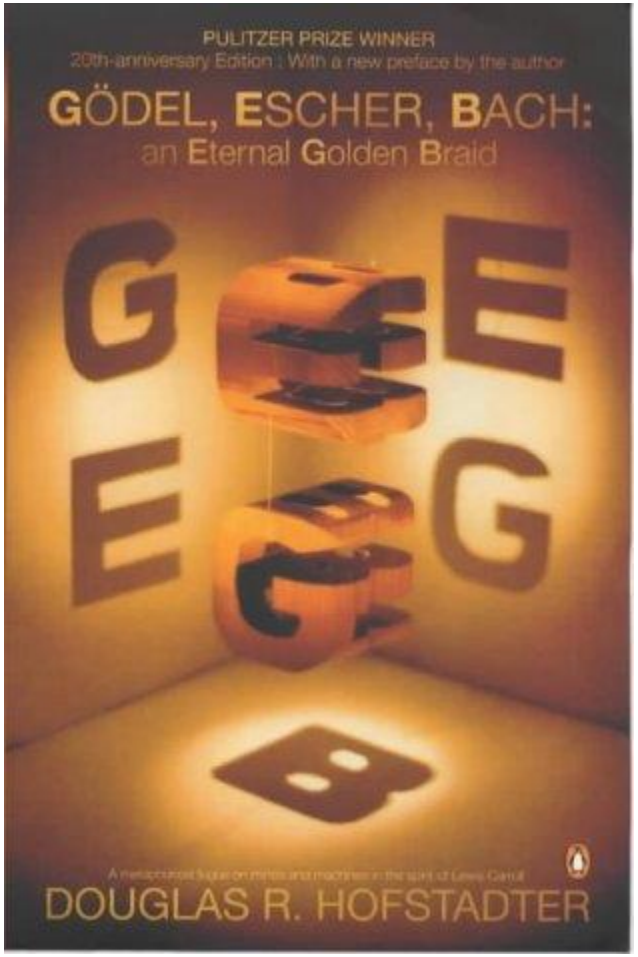
- An entertaining and inspirational introduction to the mathematical foundations of our subject - computer science.

Course Book

"Gödel, Escher, Bach: an Eternal Golden Braid"
by
Douglas R. Hofstadter

Publisher: Penguin Books Ltd
ISBN: 0140289208
Cost: £18.99 (amazon.co.uk)

You will need your own copy of this book as we will be reading from it every week.



University of Strathclyde Science

Why Gödel, Escher, Bach?

- The book covers, in an eccentric and accessible way, many of the ideas which you will encounter over and over again in computer science.
- Examples: proof, formal systems, decidability, syntax and semantics, recursion, formal languages, abstract machines, artificial intelligence, and more besides.
- The central topic of the book is Gödel's fundamental result on the incomplete nature of formal systems.
- It also has pictures by Escher (and stuff about Bach).

University of Strathclyde Science

What is Computation?

- Computation is what computers do.
- A *computer* is a *machine* for manipulating *input data* according to a list of instructions known as a *program*.
- If we can change the program (e.g. by using a text editor), then the computer is called *programmable*.
- Programs state the sequence of operations to be performed on the input data to form the *output data*.
- Programs have to be written in a formal language.

University of Strathclyde Science

Example Program

- Here is a program written in a simple programming language called MARIE:

```
LOAD A
ADD B
STORE C
HALT
A, DEC 2
B, DEC 2
C, DEC 0
```

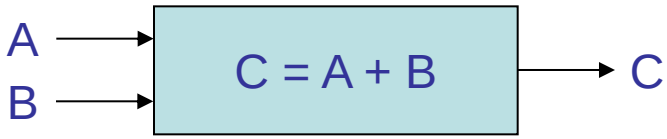
University of Strathclyde Science

Example Program

- Here is a program written in a simple programming language called MARIE:

```
LOAD  A
ADD  B
STORE  C
HALT
A, DEC  2
B, DEC  2
C, DEC  0
```

Function box:



Programs are *functions* which turn input data into output data

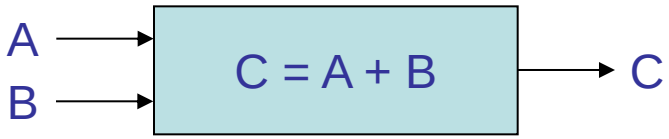
University of Strathclyde Science

Writing Function Boxes in Python

- Some languages, such as Python, allow us to write our function boxes in a more natural way:

```
LOAD A
ADD B
STORE C
HALT
A, DEC 2
B, DEC 2
C, DEC 0
```

Function box:

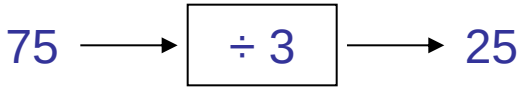



```
def add (a,b):
    c = a+b
    return c
```

University of Strathclyde Science

Function Machines

- Function machines are simple mathematical devices for transforming inputs into outputs:



- Computers are programmable function machines.
- The inputs and outputs can be numbers, text, pictures, sounds, or anything you like really...
- … but it all gets turned into binary integers anyway.

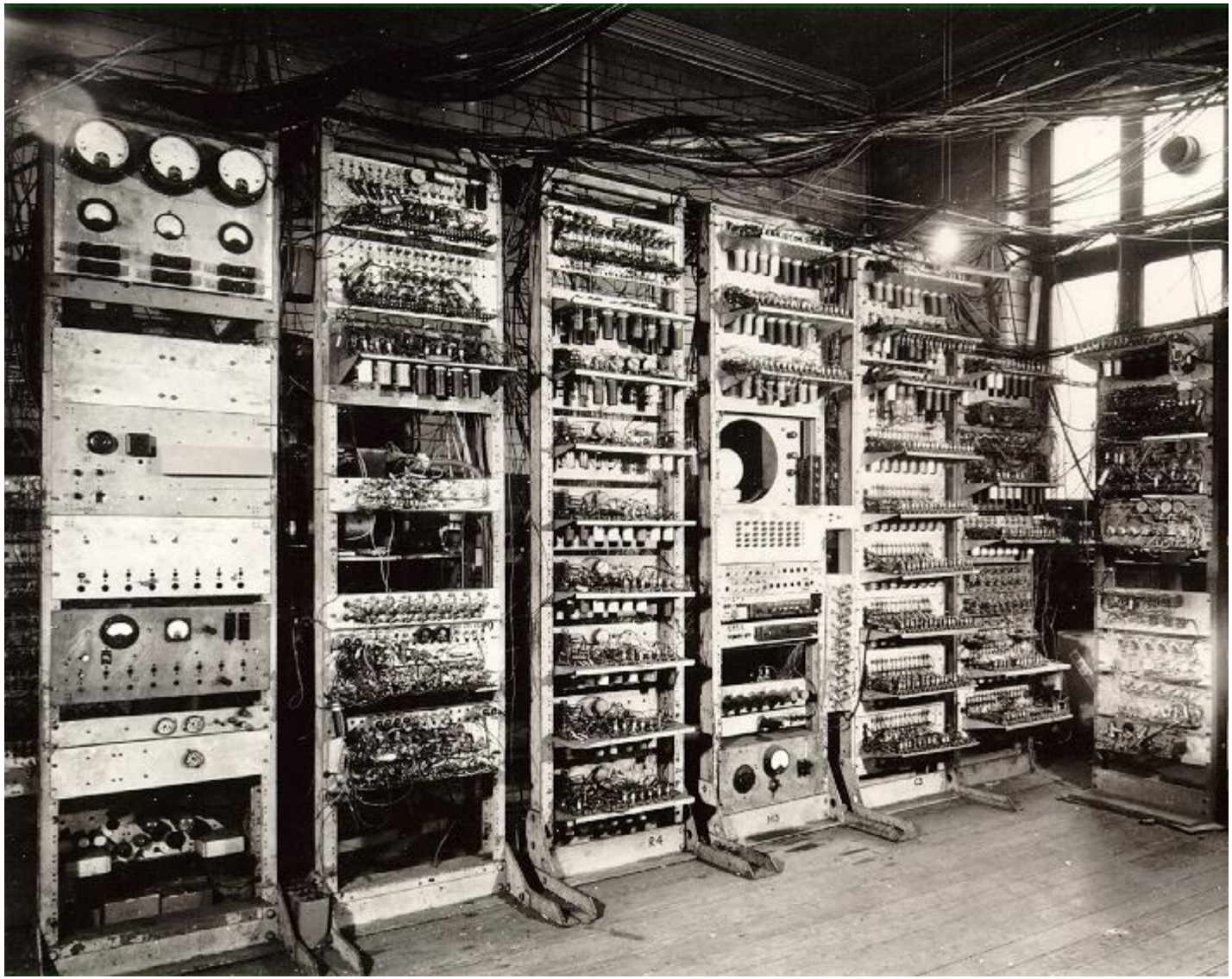
University of Strathclyde Science

Solving Problems

- We use these programmable function boxes to solve *problems* for us.
- Example problem: find the best route between two points on a map.
- The input data are the map (represented in a way that the computer understands it) and the two points.
- The output is a string of human-understandable text which gives details of the route to be taken.

University of Strathclyde
Science

Mark I Computer, Manchester, 1948



University of Strathclyde
Science

PDP-7, Digital Equipment Corporation, 1965

University of Strathclyde
Science

Cray 1 Supercomputer, Cray Research, 1975

University of Strathclyde
Science

Tianhe-2, China, 2015: The World's Fastest Computer

All Computers are Equivalent

- In mathematical terms, all computers are equivalent (in the same way that all cars are equivalent).
- If you have a problem which can be solved on one computer, then it can be solved on *all* of them! This fundamental idea is called *The Church-Turing Thesis.*
- There are some problems which cannot be solved on *any* computer, no matter how powerful the computer is.
- The mathematical properties of *all* computers can be exactly modelled by very simple abstract devices.

University of Strathclyde Science

Weeks 1 to Week 6: Course Content

- The first 6 weeks of the course will build up towards the proof that non-computable functions exist:

  If P is the set of all computer programs and F is the set of all functions $f: n \rightarrow m$ such that n,m are members of the set of natural numbers N, then $|F| > |P|$ and therefore there are some functions in F for which no computer program can exist.

- Next lecture: a first look at proof and trying to find out if there is a biggest prime number or not…