Python Programming

John Dunnion

School of Computer Science
University College Dublin

Outline

Course Outline

- Introduction/Overview
- Problem-solving and algorithms
- Computer Programming
- Programming languages
- The structure of a computer
- Python
  - Syntax
  - Semantics
  - Variables
  - Statements
  - Flow of Control
  - Input/Output
  - Arrays
  - Subprograms
- Lots and lots of examples
- …

Learning Outcomes

On successful completion of this module students should:

1. Be familiar with the important topics in computer programming
2. Understand the fundamental elements of a programming language, including variables, assignment, conditional statements, loops, input/output, arrays, functions, etc
3. Be able to design algorithms to solve simple problems
4. Be able to write computer programs using the language elements in Python to implement algorithms
5. Be able to successfully run Python programs
6. Be able to evaluate programs to find errors

A Selection of Literature

- There is no particular text-book for this module.
- There are lots of books
- There is lots of material on the Web
- However, there are some books that I will use. . .
    - John V Guttag:
      Introduction to Computation and Programming using Python (Revised and Expanded Edition)
      The MIT Press, Cambridge, Massachusetts, USA, 2013.
    - Mark Lutz:
      Learning Python (5th Edition)
      O'Reilly Media, Sebastopol, California, USA, 2013.

What is Programming?

Programming involves:

- Identifying the problem
- Designing a solution
- Expressing the solution as an algorithm
- Writing the algorithm as a program
- Translating the program into machine code
- Running (“executing”) the program
- Testing and “debugging” the program
- Putting the program into service

Why program a computer?

- Speed
- Accuracy
- Precision
- Boredom
- Danger
- Computers are better at certain tasks than humans
- . . .

Programming

- A computer program is written to solve a particular problem
- It is the programmer who solves the problem, not the computer
- The programmer gives the computer precise instructions on what to do: this is the program
- The program is a description to the computer of what it has to do and how to do it
- The program is the solution to the problem: when it is executed, we will get the required result
- The set of steps required to solve a problem is called the algorithm
- An algorithm written in a particular programming language is called a computer program

Q: What’s wrong with Natural Languages?

- Natural language is expressive, flexible, rich
    - But ambiguous and imprecise
- There are many natural languages. . .
    - But we would prefer to have one precise language

A: Ambiguity is rife in Natural Languages

For example:

- Jane claims that Sarah saw her duck
    - Whose duck ? Jane’s or Sarah’s ?

- She called me last night
    - Who is “she”

A: Natural Languages are imprecise

For example:

- Albert is tall.
- The Albert Hall in London is tall.
- The Burj Khalifa in Dubai is tall.
- The ferry is huge.
- The computer is fast.
- The hundred metres race was fast.

What is an Algorithm?

- A set of instructions that, when executed, will solve a particular problem.
- Finite set of instructions
- Runs in finite time, ie it stops eventually
- Persian mathematician, Al-Khwārizmī wrote *On the Calculation with Hindu Numerals* (circa 825 AD)
- Translated into Latin as *Algoritmi de numero Indorum* (“Al-Khwārizmī on the Hindu Art of Reckoning”)

A recipe!

- A common real-world example (or approximation) of an algorithm is a cooking recipe!
- Recipe for Tea Brack (see `http://odlums.ie/recipes/tea-brack`)
- Ingredients
    - 225g Self-Raising Flour
    - 375g packet of Fruit Mix
    - 300ml cold Tea
    - 125g Caster Sugar
    - 1 Egg (beaten)
    - Good pinch Mixed Spice

A recipe! (continued)

- Method
  1. Place fruit and tea in bowl and leave to soak overnight.
  2. Add sugar, egg, flour and mixed spice and mix well.
  3. Transfer to a greased and base-lined 900g loaf tin or a 20cm round cake tin.
  4. Bake in a pre-heated oven (170°C/Gas Mark 3) for approximately one hour or until risen and firm to the touch.
  5. Cool on a wire tray. When cold, wrap in greaseproof paper and keep for two days before cutting.

Recipes

- A recipe is not really an algorithm, because. . .
- Imprecise
  - Lots of detail left out
  - How do you beat an egg?
  - What kind of tea?
  - What is a “good pinch”
  - Which shelf in the oven?
  - What is “overnight”?
  - “Approximately” one hour? “Risen”? “Firm to the touch”?
- Ambiguous
  - “Add sugar, egg, flour and mixed spice. . . ” Add to what?
  - Fan-assisted oven?
- Take it out of the oven!

Algorithms

An algorithm is a finite set of basic instructions, which, when executed, solve a problem.

- An algorithm should be precise
- An algorithm should be unambiguous
- An algorithm (normally) takes a defined set of inputs (in a particular format)
- An algorithm (normally) produces a defined set of outputs (in a particular format)
- An algorithm should terminate after a finite length of time
- An algorithm should guarantee to produce a correct result