Lecture 2: Intended learning outcomes

By the end of this lecture, you will be able to:

- identify standard elements of a programming language;
- describe the process of writing a computer program;
- use this process to find factors by trial division.

2 Computers, languages and programming

2.1 Programs

A program is a sequence of instructions for the computer to follow. Let's think about telling someone how to cross the road.

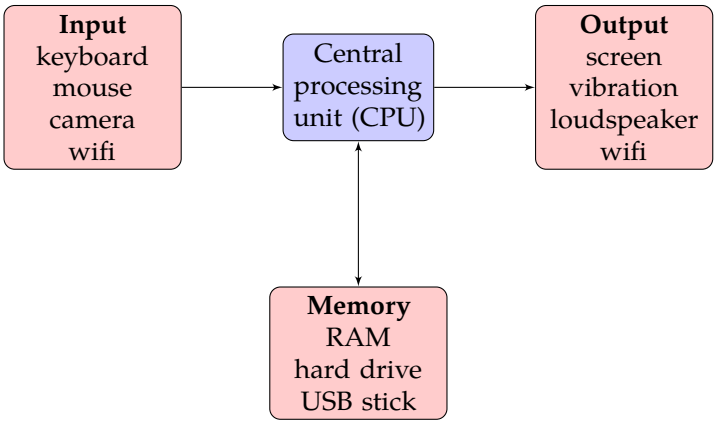
1. If there is a crossing nearby use it!
2. Otherwise, look left and right until no traffic is coming.
3. Then walk across the road, looking and listening until you get to the other side.

These instructions are written in "natural language" – in this case English. A computer program is similar but the language is more formalized. The above illustrates basic features shared by many computer languages.

2.2 Standard elements of a programming language

Variables	to store data and information, can be of various *types*	`a = 7`
Mathematical operations	add, multiply, *etc.*	`+`, `*`
Input and output operations	to communicate with the user and the World	`input`, `print`
Loops	for repeating a set of steps	`while`, `for`
Conditional statements	for checking if you want to do one thing or another	`if`, `else`
Functions/ procedures	to encapsulate a sequence of steps you perform often	`def`

2.3 Simplified schematic diagram of a computer



2.4 The process of writing a program

1. Understand the problem or task.
2. Design your solution. (In something like English.)
3. Write your program. (In Python.)
4. Test and debug your program.

Writing in Python does not come until step 3! Often you will go back to previous steps. Often you will program a little bit at a time. (Break the task down.)

2.5 A programming task

Task. For each number from 1 to 50,000 find its factors by trial division.

[*Trial division* means that to find if x is a factor of n you try to divide n by x and see if you have any remainder.]

2.6 Part 1: Understand the problem or task

1. What is the task asking me to do?
2. Do I understand what a factor is?
3. Do I understand what trial division means?

2.7 Part 2: Design your solution

1. Write down the mathematical problem.
2. Start to think how to address it algorithmically.
3. How can I perform trial division?
4. Will I need to repeat any actions?
5. Do I need a loop?
6. What do I need to print to screen?

2.8 Part 3: Write your program

```
1  n = 1
2  while n <= 50000:
3      print("The factors of", n, "are:")
4      i = 1
5      while i <= n:
6          if n % i == 0:
7              print(i)
8          i = i + 1
9      n = n + 1
```

Line 2 starts a *loop* that will repeat lines 3–9 from `n==1` until `n==50001`.
Line 5 starts a *loop* that will repeat lines 6–8 from `i==1` until `i == n+1`.
Line 7 prints `i` to screen *if* `i` is a factor of `n`.

2.9 Part 4: Test and debug your program

1. Does the program run?
2. Does the program print anything to screen?
3. Does the program print 1 and n as factors for each value of n?
4. Does the program give the correct answer for a smaller value than 50,000?
5. Does the program take a long time to run?

(We will discuss refactoring and debugging in the next two lectures.)

2.10 Conditional statements: the `if` command

```
if (boolean_condition):
    do this
    followed by this
carry on here
```

For example

```
password = input("Enter password: ")
if password == "parrot":
    print("Correct")
print("Thank you")
```