U30299 – Programming
Lecture 01 – Writing Simple Programs

Matthew Poole & Nadim Bakhshov

`moodle.port.ac.uk`

School of Computing
University of Portsmouth

2020/21

Introduction to Lecture

- This lecture introduces the basic steps involved in programming.
- We'll introduce a simple programming problem (a task to be performed), and then:
    - write down a set of steps (in English) that performs the task;
    - write these steps as a Python program; and
    - check that the program works correctly.
- We'll then study the Python program in detail to introduce the basic elements of the Python language.

A programming problem

- Consider the following programming problem:

> *Write a* weight converter program *that transforms a weight measured in kilos (kilograms) into an equivalent weight in pounds.*

Task specification

- The first step in writing any program is to make sure that the problem is understood completely.
- For the weight conversion problem, we might simply state:
  - **User input:** a weight measured in kilos
  - **Output to screen:** a weight measured in pounds, equivalent to the input weight (see next slide).

Specification — conversion formula

- The following equation relates kilograms and pounds:

$$\mathit{pounds} = 2.2 \times \mathit{kilos}$$

- For example,
    - 1 kilo = 2.2 pounds
    - 10 kilos = 22 pounds
- These two example conversions can be used later to **test** that the program operates correctly.

Designing the Algorithm

- The next step to **design** an **algorithm** that accomplishes the task.
- An algorithm is a detailed sequence of actions that accomplish some task.
- Algorithms can be written in English or any other language.
- A reasonable algorithm for our task, written in English, is:
  - Obtain a kilos value from the user
  - Calculate a pounds value using $pounds = 2.2 \times kilos$
  - Output the pounds value to the screen

The Python program

- An algorithm like this can be understood/performed by a human.
- In can also be written, or **implemented**, in any programming language like C, Python or Java.
- This algorithm is implemented as a Python program as follows:

```
kilos = float(input("Enter a weight in kilos: "))
pounds = 2.2 * kilos
print("The weight in pounds is", pounds)
```

- We'll now test the program before studying it in detail.

Testing the program

- Let's test the program to see how it looks to the user, and to check it gives correct results:

Screen

```
Enter a weight in kilos:1
The weight in pounds is 2.2
```

Screen

```
Enter a weight in kilos: 10
The weight in pounds is 22.0
```

Program concepts – statements

- Each line of the program is called a command or **statement**.
- The statements of a program are carried out (or **executed**) one after the other.
- Program execution ends after the last statement is executed.

```
kilos = float(input("Enter a weight in kilos: "))
pounds = 2.2 * kilos
print("The weight in pounds is", pounds)
```

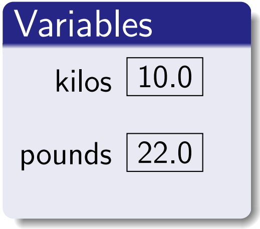
Program concepts – variables

- A **variable** denotes a part of computer memory where a value is stored.
- Variables have **names** in the program; our program has two variables, `kilos` and `pounds`:

```
kilos = float(input("Enter a weight in kilos: "))
pounds = 2.2 * kilos
print("The weight in pounds is", pounds)
```

Program concepts – variables

- We'll often draw pictures to represent the variables and their values in the computer's memory; e.g:



Variables

kilos 10.0

pounds 22.0

- A statement in the program may:
  - create a new variable
  - use the value of a variable
  - change the value of a variable

Program concepts – assignment statements

- An **assignment statement** is used to assign a value to a variable:
    - This variable appears on the left hand side of the =.
    - The right hand side is an **expression**, which has a value.

```
kilos = float(input("Enter a weight in kilos: "))
pounds = 2.2 * kilos
print("The weight in pounds is", pounds)
```

Program concepts – assignment statements

- Assignment statements are executed in two steps; they:
  1. **evaluate** the expression on the right hand side (i.e. find out its value)
  2. **assign** this value to the variable on the left hand side.
- If the variable on the left hand side doesn't yet exist, then it is **created**.
- Otherwise (if the variable already exists), its old value is replaced.

Program concepts – numeric and string values

```
kilos = float(input("Enter a weight in kilos: "))
pounds = 2.2 * kilos
print("The weight in pounds is", pounds)
```

- `2.2` is an example of a **numeric** value.
- `"Enter a weight in kilos:  "` is a **string** value.
- We can use double or single quotes, but we can’t mix them; so:
  - `"hello"` and `’hello’` are OK, whereas:
  - `"hello’` is not!

Program concepts – arithmetic expressions

```
kilos = float(input("Enter a weight in kilos: "))
pounds = 2.2 * kilos
print("The weight in pounds is", pounds)
```

- Python allows standard arithmetic expressions to be formed from +, -, * (multiplication), /, and brackets (and).
- An expression is **evaluated** to give a **value**.
- We'll study arithmetic expressions in detail next lecture.

Program concepts – built-in functions

```
kilos = float(input("Enter a weight in kilos: "))
pounds = 2.2 * kilos
print("The weight in pounds is", pounds)
```

- A **built-in function** is an algorithm that is part of the Python language, and can be accessed by using its **name**.
- The **input** built-in function:
  1. Displays a prompt on the screen;
  2. Waits for the user to enter a value;
  3. Gives us the value that the user entered.

Program concepts – built-in functions

```python
kilos = float(input("Enter a weight in kilos: "))
pounds = 2.2 * kilos
print("The weight in pounds is", pounds)
```

- We'll see more what the **float** built-in function does later.
- The **print** built-in function displays information on the screen.

Program Execution

Explanation

About to start program execution

```
> kilos = float(input("Enter a weight in kilos: "))
> pounds = 2.2 * kilos
> print("The weight in pounds is", pounds)
```

Screen

Variables

Program Execution

Explanation

```
> kilos = float(input("Enter a weight in kilos: "))
> pounds = 2.2 * kilos
> print("The weight in pounds is", pounds)
```

Screen

Variables

Program Execution

Explanation

Message is displayed; wait for the user to enter a value

```
> kilos = float(input("Enter a weight in kilos: "))
> pounds = 2.2 * kilos
> print("The weight in pounds is", pounds)
```

Screen

Variables

Program Execution

Explanation

Message is displayed; wait for the user to enter a value

```
> kilos = float(input("Enter a weight in kilos: "))
> pounds = 2.2 * kilos
> print("The weight in pounds is", pounds)
```

Screen

```
Enter a weight in kilos:
```

Variables

Program Execution

Explanation

User enters a value, let's say 10

```
> kilos = float(input("Enter a weight in kilos: "))
> pounds = 2.2 * kilos
> print("The weight in pounds is", pounds)
```

Screen

```
Enter a weight in kilos:
```

Variables

Program Execution

Explanation

User enters a value, let's say 10

```
> kilos = float(input("Enter a weight in kilos: "))
> pounds = 2.2 * kilos
> print("The weight in pounds is", pounds)
```

Screen

```
Enter a weight in kilos: 10
```

Variables

Program Execution

Explanation

A variable with name `kilos` is created, and given the value 10.0

```
> kilos = float(input("Enter a weight in kilos: "))
> pounds = 2.2 * kilos
> print("The weight in pounds is", pounds)
```

Screen

```
Enter a weight in kilos: 10
```

Variables

Program Execution

Explanation

A variable with name `kilos` is created, and given the value 10.0

```
> kilos = float(input("Enter a weight in kilos: "))
> pounds = 2.2 * kilos
> print("The weight in pounds is", pounds)
```

Screen

```
Enter a weight in kilos: 10
```

Variables

kilos $\boxed{10.0}$

Program Execution

Explanation

```
> kilos = float(input("Enter a weight in kilos: "))
> pounds = 2.2 * kilos
> print("The weight in pounds is", pounds)
```

Screen

```
Enter a weight in kilos: 10
```

Variables

kilos 10.0

Program Execution

Explanation

Expression is evaluated ...

```
> kilos = float(input("Enter a weight in kilos: "))
> pounds = 2.2 * kilos
> print("The weight in pounds is", pounds)
```

Screen

```
Enter a weight in kilos: 10
```

Variables

kilos 10.0

Program Execution

Explanation

Expression is evaluated … to 22.0

```
> kilos = float(input("Enter a weight in kilos: "))
> pounds = 2.2 * kilos
> print("The weight in pounds is", pounds)
```

Screen

```
Enter a weight in kilos: 10
```

Variables

kilos $\boxed{10.0}$

Program Execution

Explanation

A variable with name pounds is created, and given the value 22.0

```
> kilos = float(input("Enter a weight in kilos: "))
> pounds = 2.2 * kilos
> print("The weight in pounds is", pounds)
```

Screen

```
Enter a weight in kilos: 10
```

Variables

kilos 10.0

Program Execution

Explanation

A variable with name pounds is created, and given the value 22.0

```
> kilos = float(input("Enter a weight in kilos: "))
> pounds = 2.2 * kilos
> print("The weight in pounds is", pounds)
```

Screen

```
Enter a weight in kilos: 10
```

Variables

kilos	10.0
pounds	22.0

Program Execution

Explanation

```
> kilos = float(input("Enter a weight in kilos: "))
> pounds = 2.2 * kilos
> print("The weight in pounds is", pounds)
```

Screen

```
Enter a weight in kilos: 10
```

Variables

kilos	10.0
pounds	22.0

Program Execution

Explanation

A message and the value of pounds is displayed

```
> kilos = float(input("Enter a weight in kilos: "))
> pounds = 2.2 * kilos
> print("The weight in pounds is", pounds)
```

Screen

```
Enter a weight in kilos: 10
```

Variables

kilos	10.0
pounds	22.0

Program Execution

Explanation

A message and the value of pounds is displayed

```
> kilos = float(input("Enter a weight in kilos: "))
> pounds = 2.2 * kilos
> print("The weight in pounds is", pounds)
```

Screen

```
Enter a weight in kilos: 10
The weight in pounds is 22.0
```

Variables

kilos	10.0
pounds	22.0

Program Execution

Explanation

Program execution completed

```
> kilos = float(input("Enter a weight in kilos: "))
> pounds = 2.2 * kilos
> print("The weight in pounds is", pounds)
```

Screen

```
Enter a weight in kilos: 10
The weight in pounds is 22.0
```

Variables

kilos	10.0
pounds	22.0