Programming basics

Programming is writing computer code to create a program, in order to solve a problem. To program a computer, you need to know how programs are constructed.

Arithmetic

Programming **is writing computer code to create a program, to solve a problem**. Programs are created to implement algorithms. Algorithms can be represented as pseudocode or a flowchart, and programming is the translation of these into a computer program.

To tell a computer to do something, a program must be written to tell it **exactly** what to do and how to do it. In order to do this, you need to know how programs are constructed.

Basic arithmetic

Arithmetic is used all the time in computer programs, so an understanding of how computers use arithmetic is useful. This table lists the common arithmetic processes with their programming equivalents:

Arithmetic process	Programming equivalent
Addition (plus)	+
Subtraction (minus)	-
Multiplication	*
Division	/

Computers can be programmed to carry out calculations – so long as the correct formulas to use are known. Some simple mathematical examples in Python (3.x) are:

>>> print(5 + 7) 12 >>> print(7 - 5) 2 >>> print(5 * 7) 35 >>> print(35 / 7) 5.0
More complicated calculations can be performed too:

>>> print((5 * 2) + (4 - 3)) 11 >>> print((8 / 4) + (2 - 1)) 3.0

Variables

Understanding variables

Sometimes we need computers to remember the information we give it and that it calculates during programs. **A** variable **can be thought of as a box that the computer can use to store a value**. The value held in that box can change or 'vary'. A program can use as many variables as it needs it to.

All variables are made up of three parts:

- a name
- a type
- a value
-

Why use variables?

Variables are extremely useful when programming. If a variable is used for the length of a side in a simple program to calculate the perimeter of a square, it is easy to calculate the perimeter of a different sized square, simply by changing the value of the variable.

Variables are a key element of programming. They are used for calculations, for storing values for later use, in decisions and in iteration.

Naming variables

Each variable is named so it is clear which variable is being used at any time. It is important to use meaningful names for variables:

For example, pocketMoney = 20 means that the variable 'pocketMoney' is being used to store how much pocket money you have. Right now you have £20.
The name given to each variable is up to the programmer, but ideally a variable name should have meaning, ie it should reflect the value that it is holding.

Variable naming rules

There are some rules about variable names:

- Consistency: ‘name’ is not the same as ‘Name’ or ‘NAME’.
- Spacing: variable names should not have a space in them. Use underscores or camelCase instead, eg total_money; totalMoney).
- Digits: variable names should not start with a digit

Consider these example variable names, all of which could be variable names to store the length of a side of a square:

Variable name	Comment
l	A poor choice – it has no meaning
length	Okay but a bit vague
side_length	Good
sideLength	Good
side length	Wrong – don’t use spaces

Example

This Python (3.x) program uses two meaningful names when calculating the perimeter of a square:

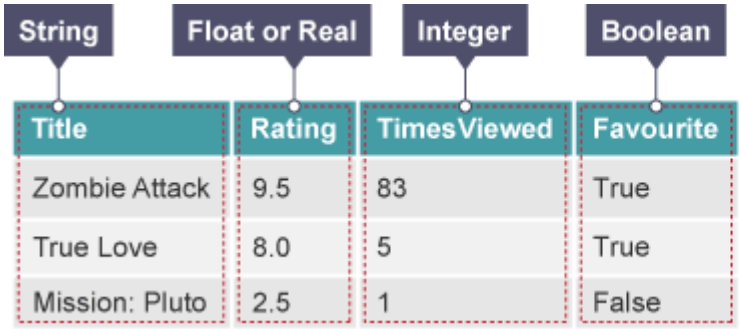
>>> side_length = 5 >>> perimeter = side_length * 4 >>> print(perimeter) 20
Because meaningful names have been used in this code, it is easy to know what each variable is used for.

Data types

Variables come in all shapes and sizes. Some are used to store numbers, some are used to store text and some are used for much more complicated types of data.

The data types to know are:

- **String** (or **str** or **text**). Used for a **combination of any characters that appear on a keyboard**, such as letters, numbers and symbols.
- **Character** (or **char**). Used for **single letters**.
- **Integer** (or **int**). Used for **whole numbers**.
- **Float** (or **Real**). Used for **numbers that contain decimal points**, or for **fractions**.
- **Boolean** (or **bool**). Used where data is restricted to **True/False** or **yes/no** options.



In many programming languages variables must be declared before they can be used, for example:

- Visual Basic - dim score as int
- Java – int score;

In some languages, such as Python, you can simply start using the variable without declaring it.