ICT Level 2 – Writing Computer Programs – Programming Terms Part 1

1 of 20 – Welcome

Welcome to this session on writing computer programs: programming terms.

By the end of this session, you will:

* Know a number of new programming terms, such as variable and declaration
* Understand what these terms mean
* Know how to use these terms when discussing computer programming

2 of 20 – Introduction to writing computer programs: programming terms

**What are programming terms?**

Computer programming is made up from a number of different techniques and processes that programmers will use, in order to build the best possible piece of software.

These techniques and processes can be thought of as programming terms, and each term relates to a different feature of programming.

When you understand these terms and techniques you will then be a step closer to building your own computer program - and closer to seeing the fun that can be had during these design stages, too!

3 of 20 – One or two programming terms

Some key terms that are usually found within computer programming are:

* Variables
* Subroutine
* Declarations
* Assignment
* Scope
* Constant

In this session we will focus on **variables** and consider what their role is within the programming process. We will also look at one or two other terms that are usually discussed alongside variables, as some of these elements have to work together in order to get their jobs done.

4 of 20 – Variables: what are they?

Variables are **memory stores** that can be used to **hold data** when a program is running.

Another way to think of variables is as small boxes where important information can be stored, and retrieved, whenever the program needs to access it.

What is kept inside this memory store – meaning the **data that has been input – can be moved at any time**, for example, if the programmer needs the data to be stored within a different variable. **The stored data can also be changed or updated** from inside the variable too.

While variables are very important in constructing a computer program, it is important to note that a program cannot actually use a variable until it has been declared.

5 of 20 – Declaration: what does this mean?

A declaration is a document of sorts, designed to hold three key pieces of information:

1. What is the name of the variable?
2. What type of data is likely to be kept inside the variable?
3. Which parts of the program have access to that variable?

Each declaration must answer these questions in a way that the software, and computer system, will be able to understand, in order for the variables to work effectively. We will look at a declaration in action in the next section.

6 of 20 – Declaration: an example

Visual Basic – which is a programming space developed by Microsoft – provides a number of declaration examples, such as:

* **Dim QuizScore As Integer**

This declaration explains that the variable titled **QuizScore** will be used to hold **whole numbers** (in the declaration, **As Integer** is the part that expresses that information).

While this is a good example of a declaration, it is important to note that **Dim is exclusively used by Visual Basic software**. Different programming languages will use different labels in place of this – so it will not always be Dim at the beginning.

7 of 20 – As Integer: what does that mean?

The phrase **As Integer**, which we saw in the previous section, refers to how th**e data inside that variable box is being stored**.

As Integer means something is being stored as whole numbers.

There are different **data types** that can be used in place of this, such as:

* Character
* String
* Real
* Boolean

We will consider these in more detail in a later session.

8 of 20 – Declarations and where you find them

Declarations can be **positioned at different points** inside a program.

Where the declaration is within the program can then be used to work out the **scope** of the variable – meaning, which parts of the program are allowed access to the variable. The scope can either be **local** or **global**.

If the scope is local, then the declaration will be placed inside the part of the program that can use the variable.

This makes the variable local to the program instructions that need to use it (this group of instructions can be called a **subroutine**), and it also means that this particular variable cannot be used in another area of the program.

9 of 20 – Declarations and where you find them – continued

The scope of a declaration can also be global. This means that the declaration exists outside of any specific instructions – or rather, any subroutines – so the declaration can be accessed by any individual part of the program, without any limitations.

Depending on what programming language is being used, there may be one or two differences to keep in mind about global variables.

The most important difference to remember is that some languages will need a global variable’s declaration statement to be featured in different parts of the program – these different parts can also be referred to as modules. This means that different parts of the program can use the variable alongside each other (rather than it being used by one part of the program at a time).

10 of 20 – Assignment statement: what does this mean?

Alongside declaration statements, programs also rely on **assignment statements** too. When a computer programmer is using assignment statements this means that they are **assigning a value to a variable** (or rather a value is being put inside the variable).

An assignment statement **can also indicate a change to an existing value** too. This means that it could be used to highlight an update to an existing variable.

Assignment statements are a little easier to express than declaration statements. This is because an assignment statement simply gives you the variable in question, and the value (or new value, if a change has been made) that belongs to that variable. This might be something like:

* **VAT = 20**

11 of 20 – Assignment statements – continued

In the example VAT = 20, VAT is the name that has been given to our imaginary variable. The value placed inside of that variable then becomes 20. This expression will be the same across various programming languages, so assignment statements will always be written as:

* **variable = value**

For example:

* Name = 50
* Height = 10

These values can then be accessed by various parts of the program.

**Remember:** the programmer can change or update these variable values whenever they need to.

12 of 20 – Constant: what does this mean?

The final new term to introduce in this session is a constant, which is quite similar to a variable, but it has one key difference.

While variables can be altered at any time during the programming process, constants cannot. Once they are input into the program, they must stay at the value that they were originally given and cannot be updated over time.

Because constants cannot be changed, they are most often used for values that are never changed in real-life settings – an example of this might be a mathematical principle, such as Pi.

Constant values – like other values – are also introduced in the declaration statement, and they can also be accessed by different parts of the program.

13 of 20 – Other elements of programming

Variables are a very important part of computer programming. Alongside variables, programmers will also have a good knowledge of:

* **Input and output** – this means how information is introduced to the program, and the effect it then has on the system
* **Subroutines** – you might remember these as small sets of code, or instructions
* **Annotations** – these are small comments made by a programmer and added straight into the code itself to explain processes, and possible changes
* **Data types** – these are the expressions used within declarations to explain the type of data involved

14 of 20 – Question 1

What is a variable?

1. A variable is a storage box where data and values can be held, so that a program can eventually gain access to it
2. A variable is a storage box inside a computer program; it can only hold a limited amount of data and the data cannot be moved

The correct answer is A, a variable is a storage box where data and values can be held, so that a program can eventually gain access to it.

15 of 20 – Question 2

Which three pieces of information can you find in a declaration?

Choose all that apply:

1. The name of the variable
2. The parts of the program that cannot access the data
3. The type of data being stored
4. The parts of the program that can access the data
5. The subroutine for that data

The correct answers are A, C and D, the name of the variable, the type of data being stored and the parts of the program that can access the data.

16 of 20 – Question 3

Match these key terms; **local** and **global**, to the definitions below:

1. The declaration, and variable, exist outside of a subroutine, which means that they are not limited and can be used by any part of the program
2. The declaration is placed inside the part of the program that has access to it; this way the variable cannot be used by other areas

The correct answers are:

The declaration is placed inside the part of the program that has access to it; this way the variable cannot be used by other areas is a definition of a **local** declaration.

The declaration, and variable, exist outside of a subroutine, which means that they are not limited and can be used by any part of the program is a definition of a **global** declaration.

17 of 20 – Question 4

Indicate whether the following statements are true or false.

An assignment statement is something that gives a value to a variable.

True

False

The correct answer is: True

An assignment statement cannot update the value of a variable once it has been input.

True

False

The correct answer is: False

An assignment statement can be expressed as variable = value.

True

False

The correct answer is: True

An assignment statement is the same as a constant value.

True

False

The correct answer is: False

18 of 20 – Question 5

Using the following choice of words; **input**, **assignment**, **declaration**, **string**, **computer** **programming** and **variable**, fill in the blanks for the paragraph below:

Variables are an important part of **blank**, but there are many other things to consider as well. For example, the **blank** and output of data alongside different data types should also be considered. Examples of data types are character and **blank**, and these are used in declarations to indicate what type of data is being stored inside the **blank**. While **blank** statements are used to express data types, **blank** statements can also be used for variables; these statements are used to attach a value to a variable, which is something that can be changed as required.

The correct paragraph should read:

Variables are an important part of **computer programming**, but there are many other things to consider as well. For example, the **input** and output of data alongside different data types should also be considered. Examples of data types are character and **string**, and these are used in declarations to indicate what type of data is being stored inside the **variable**. While **declaration** statements are used to express data types, **assignment** statements can also be used for variables; these statements are used to attach a value to a variable, which is something that can be changed as required.

19 of 20 – End

Well done. You have completed this session on writing computer programs: programming terms.

In this session we have covered:

* A number of new programming terms, such as variable and declaration
* What these terms mean
* How to use these terms when discussing computer programming

If you have any questions about any of these topics, make a note and speak to your tutor for more help.