

# FACTFILE: GCE DIGITAL TECHNOLOGY

## AS1: APPROACHES TO SYSTEMS DEVELOPMENT: PROGRAMMING



### Program Structure I: Algorithm, Syntax, Data Types and Variables

#### Learning Outcomes

Students should be able explain the terms:

- Algorithm
- Syntax
- Data Type
- Variable

#### Content in Program Structure 1 Fact File

- Explain the term algorithm
- Explain the term syntax
- What is a variable?
- Explain the term data type
- Bringing it all together

#### Explain the term Algorithm

An algorithm is a set of rules or steps that represent the solution to a problem. The algorithm is used to help design a solution to a given problem and the order of the steps is important. Given any task we can break it down into a series of steps, this makes solving the problem a lot easier.

Consider a simple problem. You have been asked to create instructions for making an omelette. The solution to this problem could be found in a recipe book. However, you might first decide to ask some questions to determine the scope of the problem. For example:

- How many eggs will be used?
- Is it a plain omelette or will other ingredients be required?

Assuming that it will be a two egg, plain omelette, the instructions could be as follows:

#### Making an omelette

1. Heat Pan
  2. Grease the pan
  3. Place two eggs in a bowl
  4. Whisk together for 2 minutes
  5. Add half a cup of milk
  6. Whisk for a further minute
  7. Add the whisked mixture to the pan
  8. Cook for three minutes
- You have a perfect omelette

- A large problem has been broken down into smaller sub problems
- The algorithm represents the solution to the problem
- Note that the order in which the steps 1–8 are carried out is important (can step 4 be carried out before step 3?)
- There may be many different ways to make a two egg plain omelette. This represents only one solution.

Algorithms can be used to help create solutions when a program is required.

A program is required to add two numbers together and output the result.

PROBLEM: ADD TWO NUMBERS AND OUTPUT RESULT

GET FIRST_NUMBER	(enter the first number)
GET SECOND_NUMBER	(enter the second number)
CALCULATE SUM = FIRST + SECOND	(add first and second giving sum)
PUT SUM	(output the result of the calculation)

The word GET is used to represent the input of information from a file or from the screen. The word PUT is used to represent the output of information to a file or onto the screen.

Question: A program is required which will take as input two numbers representing the sides of a rectangle and output the perimeter and the area of the rectangle.

To design a solution to this first break the problem down into three explicit areas:

1. INPUT *What data should be input to the program?*  
Ans : length, breadth.
2. PROCESS *What will the program do with the data?*  
Ans : Area = length\*breadth, perimeter = 2\*(length + breadth).
3. OUTPUT *What results are to be output?*  
Ans : Area, perimeter.

From this we can see that we require **4 variables** to hold data. What are these?  
Ans: length, breadth, area, perimeter.

Now we can write our algorithm:

```

PUT ENTRY MESSAGE1
GET LENGTH
PUT ENTRY MESSAGE2
GET BREADTH
CALCULATE PERIMETER=2*(LENGTH+BREADTH)
CALCULATE AREA=LENGTH*BREADTH
PUT AREA
PUT PERIMETER
  
```

Having provided a solution, a program can be written from this.

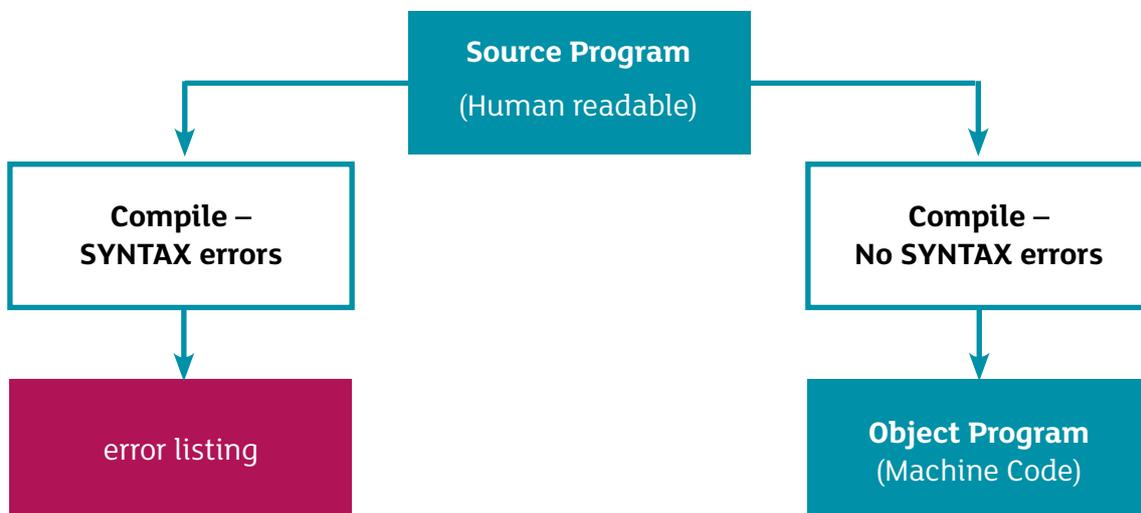
C# code for the algorithm	Output from the program
<pre> namespace AreaandPerimeter {     class Program     {         static void Main(string[] args)         {             int length, breadth, area, perimeter;              Console.WriteLine("Enter length");             length = Convert.ToInt32(Console.ReadLine());              Console.WriteLine("Enter breadth");             breadth = Convert.ToInt32(Console.ReadLine());              area = length * breadth;              perimeter = 2 * (length + breadth);              Console.WriteLine("Area = " + area);              Console.WriteLine("Perimeter = " + perimeter);              Console.ReadKey();         }     } } </pre>	 <pre> Enter length 4 Enter breadth 5 Area = 20 Perimeter = 18 </pre>

Question: Discuss how the order of the statements in the algorithm could impact on the accuracy of the programming solution.

### Explain the term *syntax*

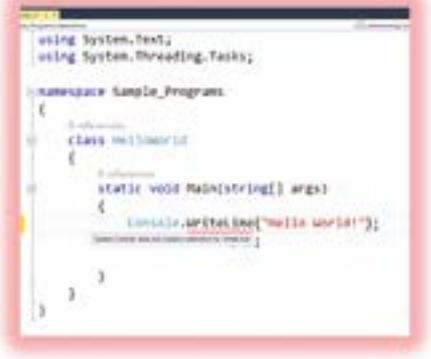
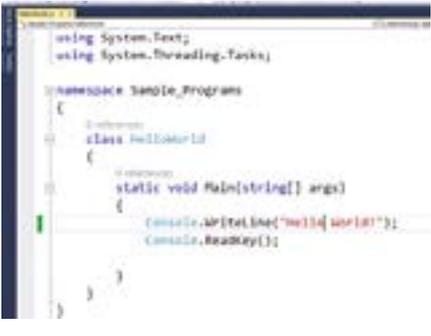
All languages have rules governing how statements should be structured. These rules are known as the **syntax** of the language. In a computer programming language, syntax rules specify how commands should be used to create statements. A statement is a single line of code which performs a task or calculation. A number of statements make up the source code. Statements can contain many aspects of the language such as: keywords, variables and assignment statements

The source code is compiled and the statements are checked to ensure that they follow the syntax rules of the language. If the statements do not follow the syntax rules, a list of *compile-time* errors will be generated, by the compiler. The programmer will have to correct all of the syntax errors before the program can be fully compiled into machine code.



For example

Here is a simple Hello World! Program in C#.

Source Code Statement	Resulting Output
 <p><code>Console.WriteLine("Hello World!");</code> Error in keyword</p> <p>Note the <i>syntax error assistance</i> aspect of the IDE described in the "Programming Environment" fact file.</p>	 <p>Error 1 'System.Console' does not contain a definition for 'WriteLine' HelloWorld.cs13 21 Sample Programs</p> <p>✗ Error listing shows syntax error</p>
 <p><code>Console.WriteLine("Hello World!");</code> Error corrected</p>	 <p>✓ Code successfully compiled and executed</p>

A simple Hello World! program in python.

Source Code Statement	Resulting Output
 <pre>1 print "Hello, World!"</pre> <p>Print "Hello, World!" Error in keyword</p>	 <p>✘ Error listing shows syntax error</p>
 <pre>1 print "Hello, World!"</pre> <p>Print "Hello, World!" Error corrected</p>	 <pre>Hello, World!</pre> <p>✔ Code successfully compiled and executed</p>

**What is a variable?**

A variable is a named location in the computer's memory which is used to hold a data value whilst the programming is running. A variable can take on different values during the course of the program execution and is accessed by referring to the variable name.

Most programming language require the programmer to declare all variables by listing their names and their data type.

**Explain the term Data Type**

When designing a solution, the programmer must decide on a name and a data type for each variable to be used.

The data types available vary between languages but in general there is a facility for storing variables with the following types of data.

Data Type	Explanation	Example	Size in memory as implemented in Visual Studio  For exemplification use only.
<b>Numeric</b>	Used to store numeric values. These include: Integers signed and unsigned Decimal numbers: fixed and floating point.	Integer : -23, 679  Decimal 3.4, 68.9  For example the number 123.456789  As an integer is: 123 As a fixed point number with 2 decimal places: 123.46  As a floating point number: 123.456789	4 or 8 bytes  8 or 16 bytes
<b>Character</b>	Stores a single unicode character	'a' 'z'	2 bytes
<b>String</b>	Stores a sequence of zero or more characters	"Hello World"	
<b>Boolean</b>	Stores values that are interpreted as True or False.	True False	
<b>Date</b>	Stores date and time information	12/12/90	8 bytes

Different data types take up different amounts of space in memory as shown above.

Consider the program which calculates the area and perimeter. There are four variables, length, breadth, area and perimeter, are all of type integer.

```
static void Main(string[] args)
{
    int length, breadth, area, perimeter;
    Console.WriteLine("Enter length");
    length = Convert.ToInt32(Console.ReadLine());
    Console.WriteLine("Enter breadth");
    breadth = Convert.ToInt32(Console.ReadLine());
    area = length * breadth;
    perimeter = 2 * (length + breadth);
    Console.WriteLine("Area = " + area);
    Console.WriteLine("Perimeter = " + perimeter);
    Console.ReadKey();
}
```

The data type for each of the four named variables used in the program is declared as int. This tells the compiler that the data should be stored as integers(whole numbers). It is good practice and necessary in most programming languages, to declare all data and to give them a data type.

## Bringing it all together

Consider the following problem: **An assistant in a supermarket earns £6.20 per hour. Regardless of the income the employee must pay 25% of the total earned in tax. Write a program which will work out tax payable. Write an algorithm which will represent a solution to this problem.**

### 1. Identify Outputs and give each variable a name.

1. taxPayable
2. netPay
3. grossPay

### 2. Identify Input and give each variable a name.

1. hoursWorked
2. rateOfPay

### 3. Identify the Process required to produce the output.

```
grossPay = hoursWorked* rateOfPay
taxPayable = 0.25 * GrossPay
netPay = GrossPay - taxPayable
```

How can this be converted to a program?

#### 4. Classify the data types

Description of data	Variable name	Data type
<b>Hours worked by employee</b>	hoursWorked	Integer
<b>Total pay earned by employee before tax</b>	GrossPay	Decimal/Float/Double
<b>Total pay after tax</b>	netPay	Decimal/Float/Double
<b>Total amount of tax payable by the employee</b>	taxPayable	Decimal/Float/Double

#### 5. Write down a formal algorithm.

```

PUT "Enter hours worked"
GET hoursWorked
PUT "Enter rate of pay"
GET rateOfPay
GrossPay = hoursWorked * rateOfPay
taxPayable = 0.25 * grossPay
netPay = grossPay - taxPayable
PUT grossPay
PUT netPay
PUT taxPayable

```

#### 6. Convert this algorithm to a program in a language of your choice.



- 2 a) Explain the term variable in the context of a computer program. [3]

.....

.....

.....

- b) Select appropriate data types for the following:

Data	Data Type
A variable to hold an employee's name	[1]
A variable to hold an employee's date of birth	[1]
A variable to hold an employee's National Insurance number	[1]
A variable to hold to hold an employee's annual wage	[1]

- 3 Syntax errors can occur in a program.

- a) Explain the term 'syntax error'. [2]

.....

.....

.....

- b) Give two examples of syntax errors which you have experienced when using a programming language. [4]

.....

.....

.....

.....

**4 Practical Task**

- a) Using the algorithms you have designed in 1a) – 1c) create the corresponding programs in a language of your choice.
- b) Keep a record of syntax errors and note what you have done to correct them.

