Lesson 2: Variables, Constants & the Assignment Statement

In a nuthsell, this lesson will cover:

- String Variables
- Constants and the Assignment Statement

String Variables

Soon, you should learn how to input text by the user using 'string variables'. The following program is written showing an example of a string variable, prompting the user to input his name:

```
Program Lesson2_Program1;
Var name, surname: String;

Begin
          write('Enter your name:');
          readln(name);
          write('Enter your surname:');
          readln(surname);
          writeln;{new line}
          writeln;{new line}
          writeln('Your full name is: ',name,' ',surname);
          readln;
End.
```

If we take a look at this program, you might have realized the introduction of a new data type: the **String** data type. Both the name and surname variables are of type string. When the program is run and prompts the user to input his name, the name which is keyed in by the user goes directly to its data placeholder in memory called **'name'**. The same occurs to surname. Be reminded that the variables names **'name'** and **'surname'** are not called reserved words, but are used by the programmer as variable identifiers. I could have used **'n'** instead of **'name'** and similarly **'sname'** instead of **'surname'** but one should always assign a meaningful name to a variable. The two 'empty' writeln's in lines 9 and 10 are used to move the cursor further down on to a new line. In this case, 2 lines are cleared. The next message displays the full name of the user using the above format. If a string variable is required to be displayed on screen, it should be put concatenated with the rest of the text using a comma.

An example of this is as follows: (note where you should put the single quotes and commas)

```
Writeln('Your name is: ',name);
```

or:

```
Writeln('Your name is:',name,'. Your surname is ',surname,'.');
```

You can even make it this way (but this is not required since the text has ended):

```
Writeln('Your name is: ',name,'');
```

BUT you should always put the inverted commas properly (underlined) to close the string in the last *writeln* function.

Back To Top ⇧

Constants and the Assignment Statement

Apart from variables, there are also items in the program which are referred to as 'constants'. Unlike variables, constants keep their value or string unchanged for the whole program. Here I have written a program, quite similar to the previous one, emphasizing the use of **constants**:

```
Program Lesson2_Program2;
Var
          surname: String;

Const {the reserved word 'const' is used to initialize constants}
          name = 'Victor';

Begin
          Write('Enter your surname:');
          readln(surname);
          writeln;
          writeln;
          Writeln('Your full name is: ',name,' ',surname);
          Readln;
End.
```

In the above program, the constant 'name' is assigned to the string value of 'Victor' and is of type string. However, in other cases, you might want to use integer constants (whole numbers), or any other data types i.e.:

```
Const
          age = 15;
```

The constant 'age' is a value that could be used wherever it is required in a program.

Example:

```
age2 := 15;
age2 := age + 15;
```

The above example shows an addition of the value of the variable 'age' plus the value of 15. The value of the constant 'age' remains 15, but the value of the variable 'age2' becomes 30. The assignment statement is not only used for additions, but is also used to assign a variable: text if it is a string variable or a numeric value if it is an integer variable. Think about the following:

```
name := 'victor';
```

```
age := 15; {also: "age:='15';" BUT in this case, 'age' is an integer variable i.e. a whole
number}
writeln('Name:',name,'. Age:',age,'.');
```

Lesson 2 will be concluded with another simple program - basically a program which is mostly inputs and outputs or prompts (i.e. the computer asks the user for data input); and finally ends with an output of the computation of the total cost of fuel consumed based on the distance travelled in kilometers every week.

```
Program lesson2_Program3;
Var PD, Dname, Cmodel : String;
    TotalKM, CostPD, TCostPD, Distance : Real;
     {real is a decimal (described later)}


Begin
          TCostPD := 0; { note that this is called an 'initialisation'.
                                          It is important to initialise integer variables to
0 so that
                                          they are 'cleaned' from previous 'trash' values in
memory. }
          Writeln('This program prompts you to ' +
                    + 'input the cost per litre of');
          Writeln('the petrol/diesel you spend ' +
                    + 'in and the average distance you travel');
          Writeln('with your car every week. Then ' +
                    + 'the computer calculates the total cost');
          Writeln('you spend in fuel every week.');
          Readln;
          Write('Diesel or Petrol?: ');
          Readln(PD);
          Write('Name Of Driver: ');
          Readln(Dname);
          Write('Car Model: ');
          Readln(Cmodel);
          Write('Cost of Diesel/Petrol: (£) ');
          Readln(CostPD);
          Writeln('Average distance you travel ' +
                    + 'with your car every week: (kilometres) ');
          Readln(Distance);
          Writeln;
          Writeln;
          Writeln('Name of Driver:',Dname);
          Writeln('Car Model:',Cmodel);
          Writeln('Diesel/Petrol:',PD);
          Writeln('Average distance covered '+
                    +'every week: ',Distance:1:2,'Km');
          Writeln('Cost of ',PD,' per liter: £',CostPD:1:2,'/litre');
          Writeln;
          Writeln;
          TCostPD := Distance * CostPD;
          Writeln('Total cost of ',PD,' per week:' +
                    + '£',TCostPD:1:2); {note this,}
          TCostPD := 0;
          Writeln('Total cost of ',PD,' per week:'+
                    + '£',(Distance * CostPD):1:2); {this}
          Writeln('Total cost of ',PD,' per week:'+
                    + '£',Distance * CostPD); {and this - without ':1:2'}
          Readln;
End.
```