PRG1	**Programming I** Diploma in IT / FI / ISF Year 1 (2018/19) Semester 1	Week **2**
		1 hour
	Mission 2-2: Program Errors & Debugging	

OBJECTIVES

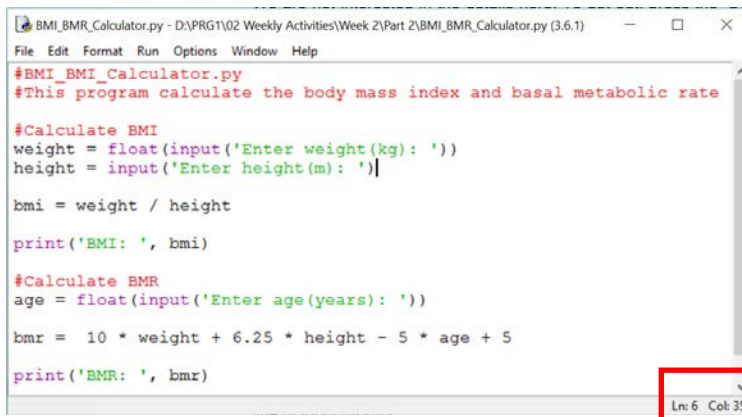At the end of this exercise, students should be able to:

- Understand the various types of program errors

- Understand the process of debugging

Part A. Introduction to the Debugger Tool

You will now learn how to use the **IDLE Debugger** to execute a program one statement at a time and examine the changes in the variable values.

Follow these instructions carefully:

i. Download the file **Mission22-BMI_BMR_Calculator.py** from Coursemology.
ii. Open the file using IDLE.
iii. The display at the bottom right tells you the current line and position in the line.
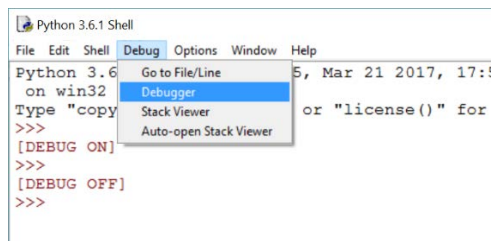
iv. Enable the debugger

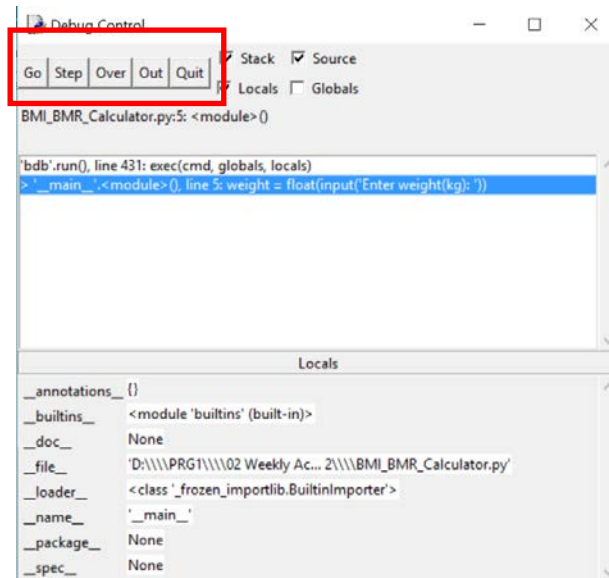Click the Debug>Debugger menu in the Python Shell to make the *Debug control window* appear.

The Debug control window:
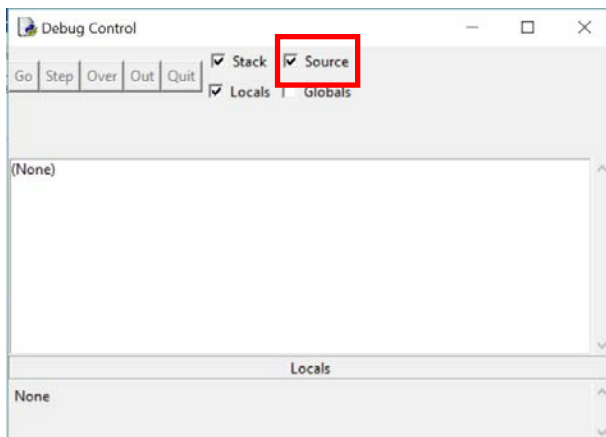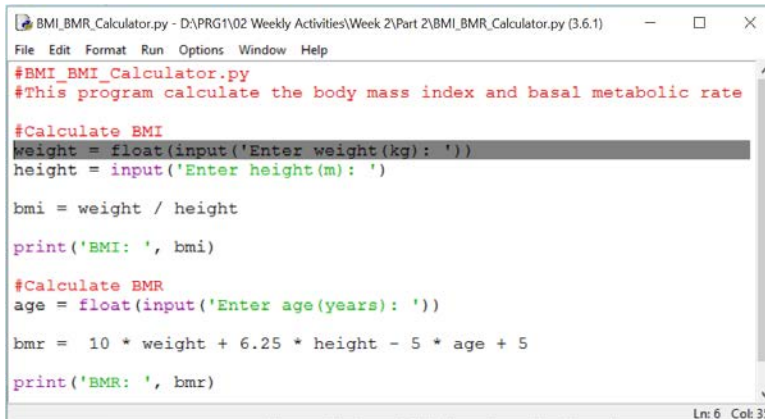

Check the 'Source' checkbox as shown above.

v.   Start the debugging process

Run the program by pressing F5, the *debugger* will be activated.

Notice that some buttons at top left hand corner in the Debug window are now enabled: Go, Step, Over, Out, Quit.

Looking at the program, you will see a line highlighted in grey. This is the next instruction that will be executed.
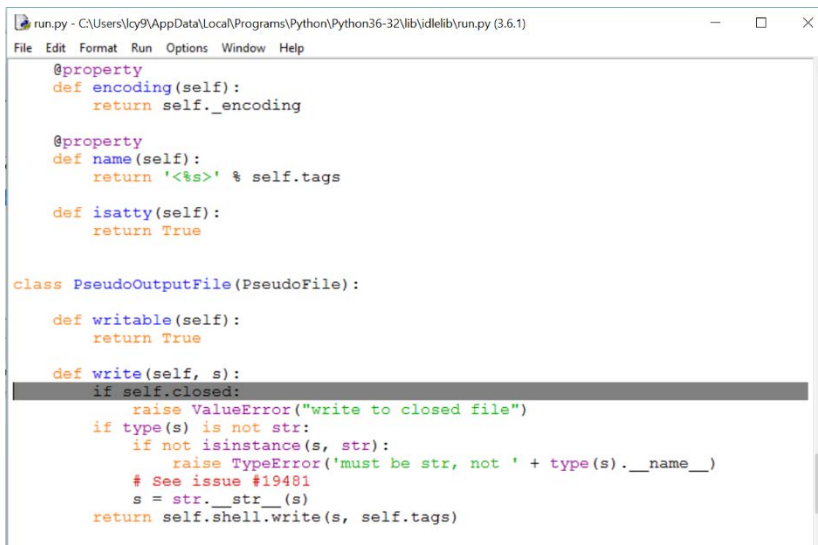
vi. Step through the program

Press the 'Step' button in Debug window to single step through the program.

When the 'Step' button is pressed, notice that the program control will jump into a function (to be learnt in subsequent topics).
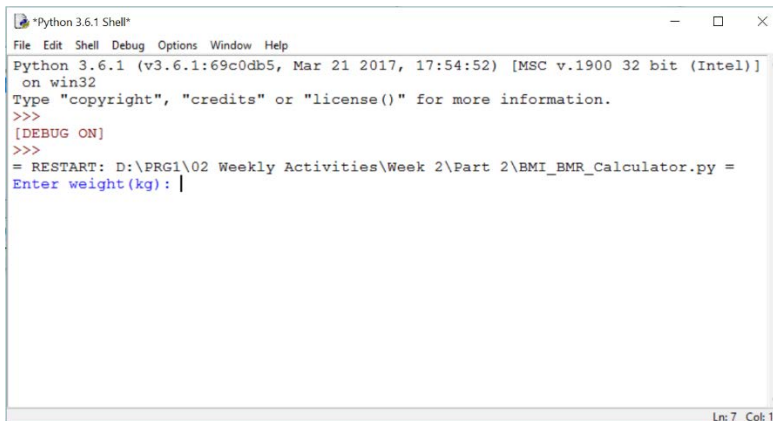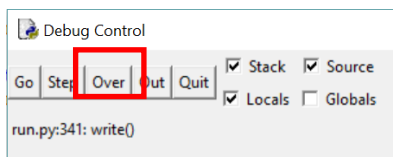

We are not interested in the details here. To get out, press the 'Out' button in Debug window, after which the input prompt for weight is shown in Python shell.
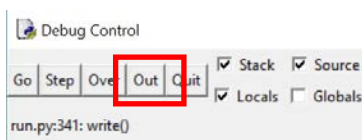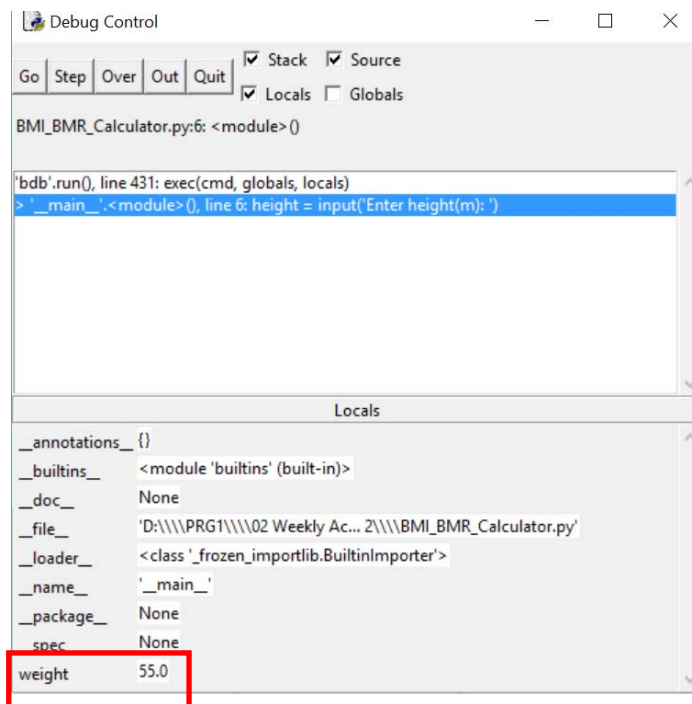
Alternatively, if you know most likely that line of code is not the cause of problem and you do not wish to see the details, press 'Over' button before executing the line of input prompt.

Enter the value for the weight as 55 and press Enter. Notice the changes in the Debug window. Weight is now 55.
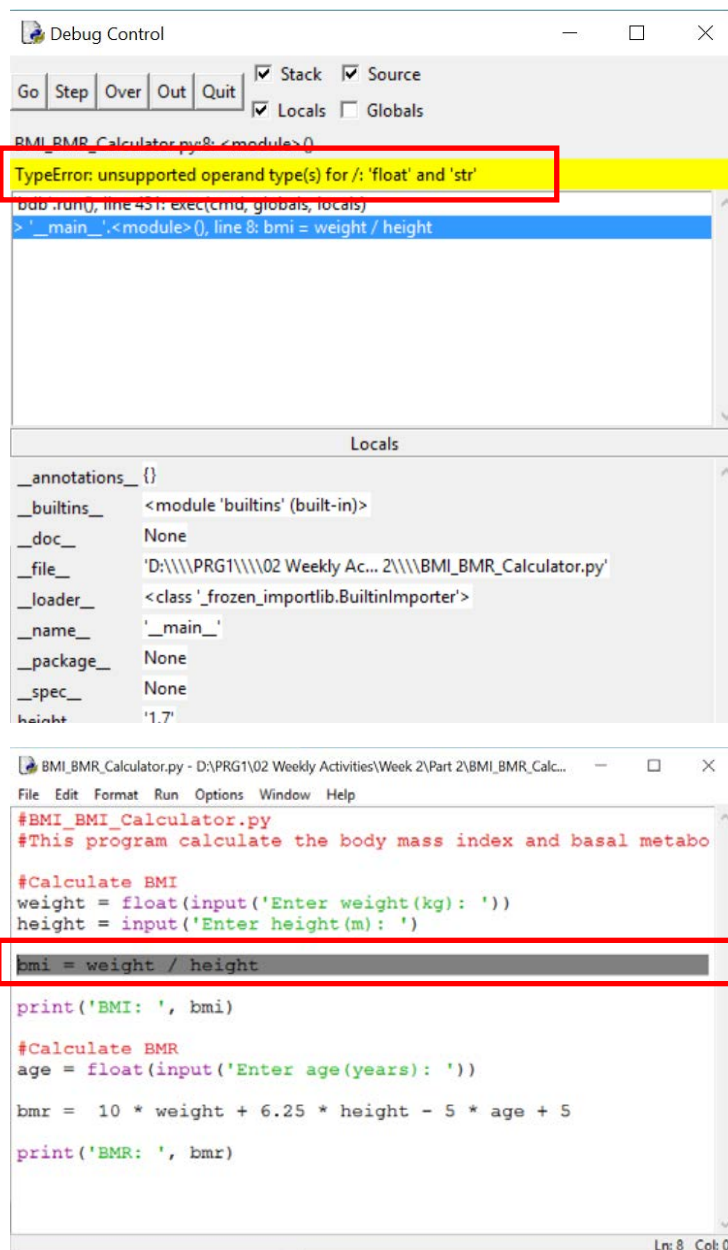
Continue to step through program, use 'Out' or 'Over' if necessary and enter the height:

Upon pressing 'Step' button further, the debugger produces the error and the statement that produced the error can be identified via the line highlighted in grey.
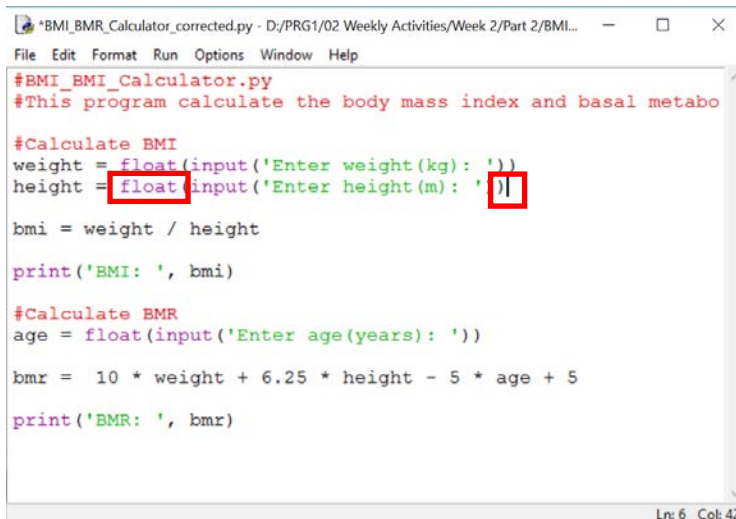
vii. Stop the debugger

Press the 'Quit' button to terminate the debugging process.

viii. Correct the error and run the debugger again

Correct the above error in the program:
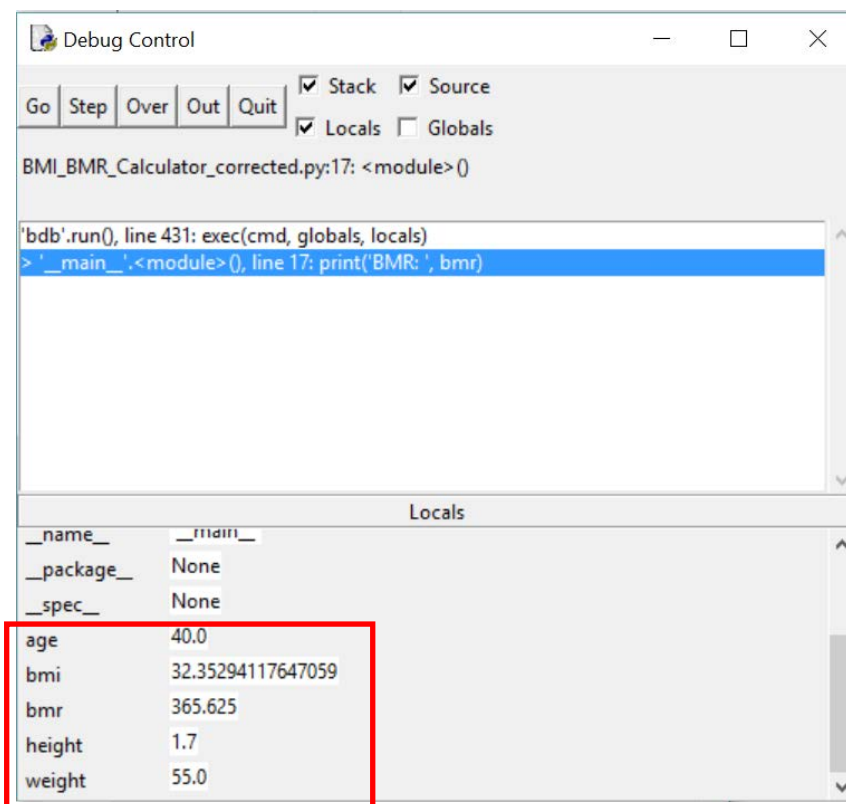
What you have just done is to correct a *syntax error*. Now run the debugger again with the corrected program. Continue to step and enter the age as 40. Inspect the changes in the variables weight, height, age, bmi and bmr. Notice how the values of variables will appear in the debug window:
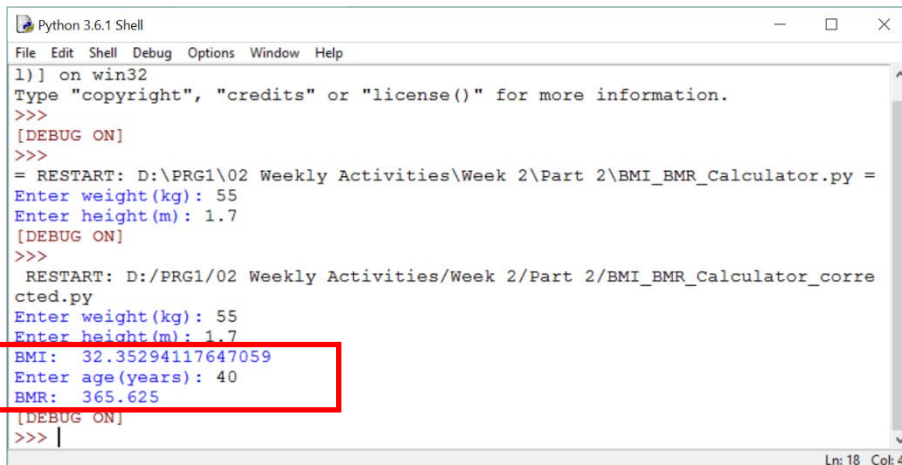
Step the program until all statements are executed. You should see the following output:
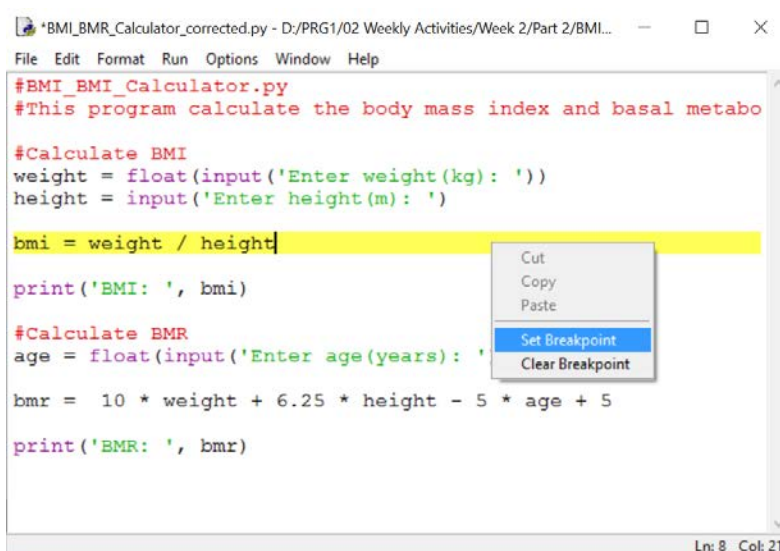
ix. Set **breakpoints** in the program.

Sometimes, we do not want to start debugging from the first line of the program, especially if you have a vague idea of which statement causes the problem. You can choose to run the program till a certain point in the code before you do more detailed investigation. This point is known as a *breakpoint*.

Now right-click on the following line in the file editor and select *Set Breakpoint* from the menu that appears. The line will be highlighted with yellow. Many breakpoints can be set in a program.
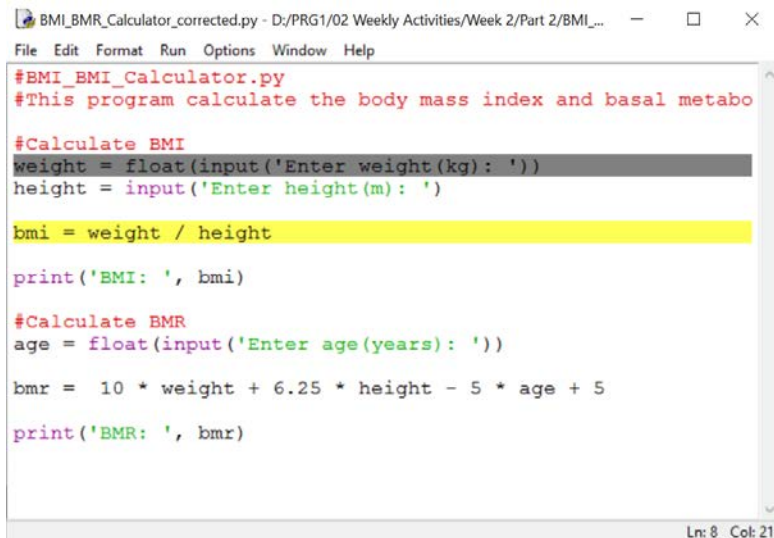
(To remove a break point, right-click on the line and select *Clear Breakpoint*.)

Now enable the debugger and run the program with F5.
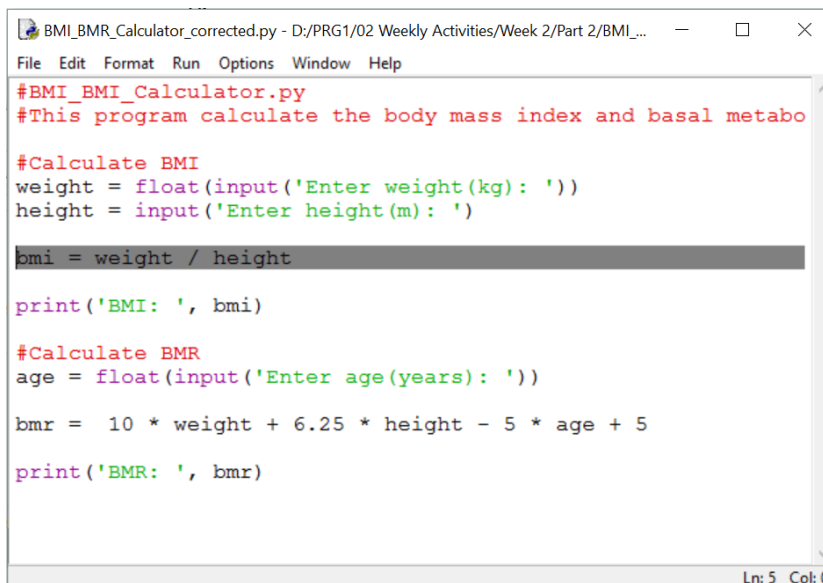
Press the 'Go' button in the debug window.
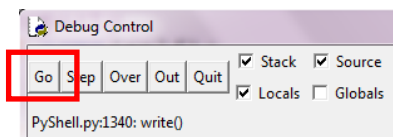
Enter the weight and height as before. You will observe that the following line is highlighted in grey. The highlighted line is now the next line to be executed. By setting a breakpoint and pressing 'Go', the program continues execution without stopping until the breakpoint.

Continue your task of stepping through the program to trace the changes in the variables.

x. Trace Table

1) Download the file **Mission22-TraceTable.docx** from Coursemology.
2) Complete the trace table in that document to record your inspection results of the variables when each statement is executed.
3) Submit your completed document to Coursemology.

xi. Observe the output of the program. It does not produce correct result although the program runs smoothly. Identify and correct the *logic errors* in the program.

Test run your code in Coursemology.

Part B. Correction of Program Errors

i. Download the file **Mission22-InterestCalculator.py** from Coursemology.

ii. Open the InterestCalculator.py using IDLE.

iii. Given the principal in dollars, rate and duration in years, the formula for calculation of interest is as follows:

$$interest = principal \times (rate \div 100) \times duration$$

iv. Correct all the errors in the program. Test run your corrected program with Coursemology.