# U08008 Modern Computing Technology Coursework 2

In this coursework, you will be assessed on learning outcomes 1, 2 and 3 as given below.

On successful completion of this module, students will be able to:

|  |
| --- |
| 1. *Design, implement and test computer programs, derived from application based case studies, implemented using a modern computer programming language.* |
| 1. *Demonstrate a thorough understanding of the fundamental concepts of high level programming languages, including the syntax and semantics of constructs for input/output, control flow and elementary data structuring.* |
| 1. *Appreciate the steps involved in translating a program description expressed in a high level language to a run time executable form.* |

In this assignment, you will be designing, coding and testing a program to store and analyse rainfall figures. You should be writing functions for each of the operations that the program will carry out but you do not need to create libraries. You should be making use of most of the programming constructs we have learned including lists and parameter passing.

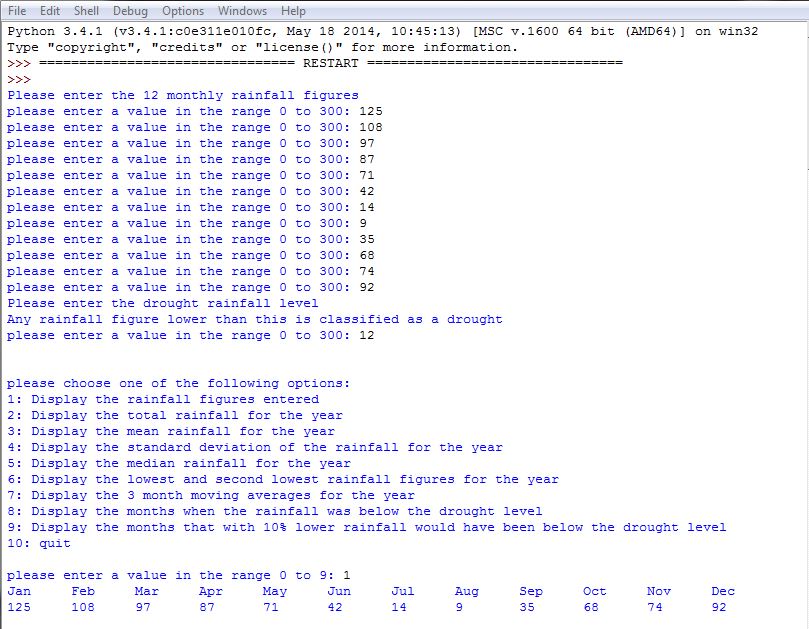
## Specification

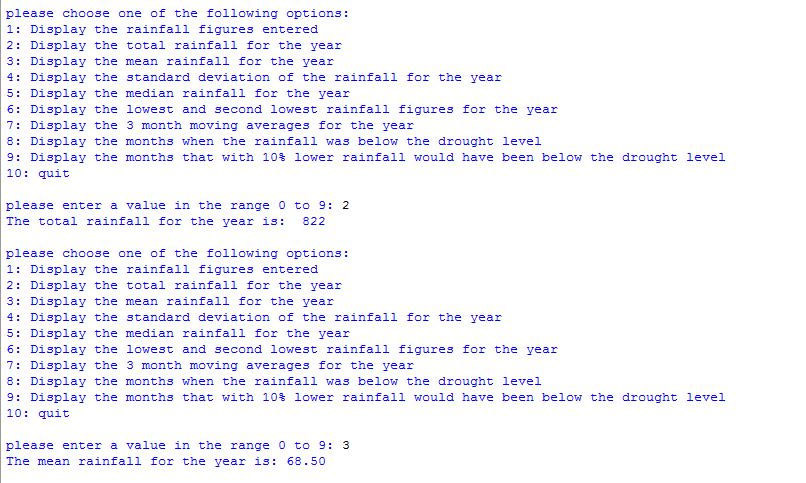
The program should ask the user for 12 monthly total rainfall figures and a drought rainfall level. The program should store these and then offer the user a series of numbered options. It will continue to do this until the user chooses to enter the number 0 to quit.

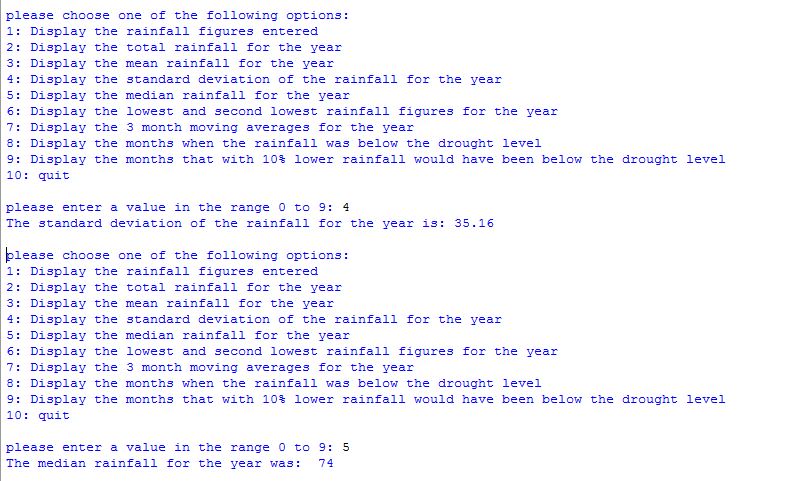
Your program should offer the following options

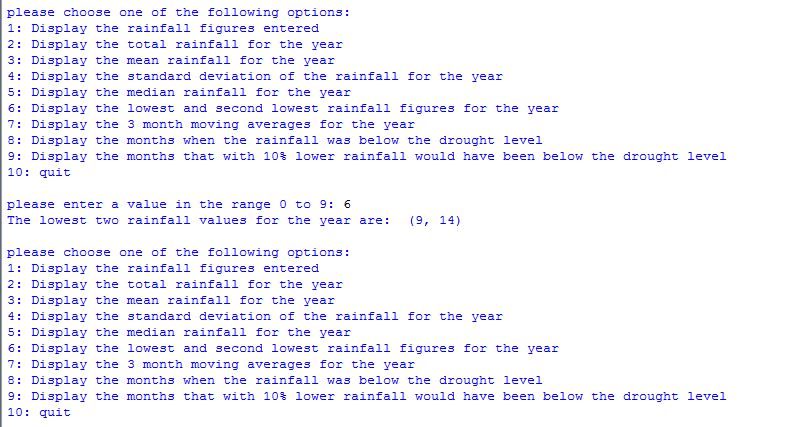
1. Quit
2. Display the data entered
3. Work out and display the total rainfall for the year
4. Work out and display the mean rainfall for the year
5. Work out and display the standard deviation of the rainfall for the year.
6. Work out and display the median rainfall for the year
7. Work out and display the lowest and second lowest rainfall for the year.
8. Work out and display the 3 month moving mean rainfalls for the year
9. Work out and display the months when rainfall was below the drought level
10. Work out and display the months when 20% lower rainfall would have put a month below the drought level.

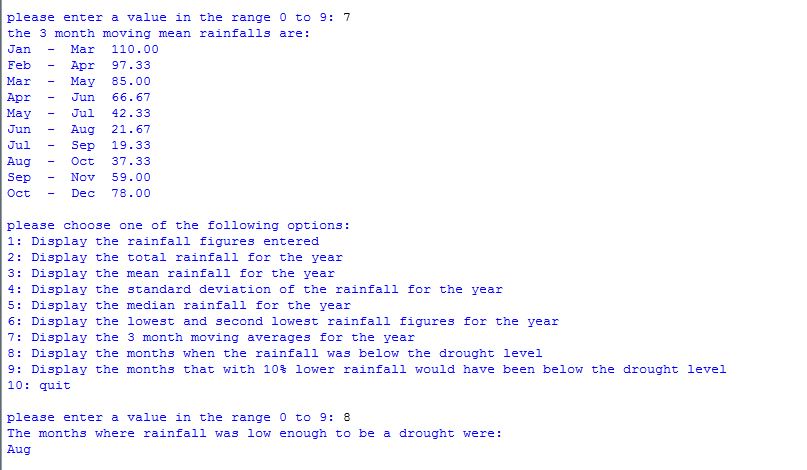
The program should only expect integer values to be entered. Any figures from a calculation that could be displayed as floats e.g. the mean, moving means or the standard deviation should be displayed to two decimal places. Data validation should simply check that rainfall figures are in the range 0.0 – 300.0 mm and choices are in the range 0 – 9. Where out of range values are entered, the program should go into a loop asking for a new value to be entered until a valid value has been entered. The screenshots show the operation of a program that meets the specification. NOTE: The rain fall figures and drought level given are not actual values for any country, they are random values used for testing[[1]](#footnote-1).











## 

## What you need to do

You should produce a main program and at least the following functions. Your code may be more efficient if you also produce additional functions to help such as for example a function that returns the month name for a given month number. You should make effective use of what you have been taught so far particularly lists and make extensive use of functions and use parameter passing. You do not need to create libraries. Only one python file is expected. The specifications below do not generally say where to use parameter passing and only list a minimum set of functions to produce. Using more functions and more effective parameter passing may be more efficient and if so will be rewarded.

### General Input Output and Data Validation

menu – this is a function should display the available choices on the screen

getValue This function should ask the user for and then return a whole number numeric value that has been entered by the user. When a value is entered by the user, if it is not in the required range, the function should keep asking the user for another value until they do enter one that is in range. HINT: parameters for the low and high limits to check against will allow the one getValue function to be used for all data entry.

enterRain – this function should read in the rainfall values

enterDroughtValue – this function should read in the drought level rainfall value

### Initial data display and calculations

displayData – this function should display the values that have been entered under month name headings. To make this easier, use three letter names for the months, e.g. Jan and assume that a full set of 12 values will fit on one line.

calculateTotal – this function should return the sum of the entered values

calculateMean – this function should return the mean of the entered values

movingMean – this function should return a set of 3 month mean averages for the year, e.g. mean of January to March, mean of February to April, mean of March to May etc.

### Standard Deviation

calculateStdDev – this function should return the standard deviation of the rainfall figures entered.

The standard deviation is the square root of the mean of the differences of each data value from the overall mean squared.

e.g.

2, 4, 5, 6, 8 have a mean of 5.

The squared differences of each value from the mean are:

(2-5)2 = (-3)2 =9

(4-5)2 = (-1)2 = 1

(5-5)2 = (0)2 = 0

(6-5)2 = (1)2 = 1

(8-5)2 = (3)2 = 9

The mean of these differences = (9+1+0+1+9)/5 = 4

The standard deviation = √4 = 2

### Calculations relating to ordering

calculateMedian – this function should return the middle rainfall value if all the values were placed in order from lowest to highest. For an even number of values, it should return the lower placed of the two middle ones.

lowestTwoValues – this function should return the lowest two rainfall figures entered.

### Drought Figures

droughtMonths – this function should return the names of the months where the rainfall figures were below the drought rainfall figure entered.

nearDroughtMonths – this function should return the names of the months where either the rainfall was already below the drought figure or would have been below the drought figure if the rainfall had been 20% lower.

### Main Program

The main part of the program should first ask the user for the 12 months’ rainfall figures and the drought rainfall figures. It should then go into a loop offering the user the 10 choices and carrying them out until the user enters 10 to quit when it should provide a goodbye message and end. All data entry should check that the numbers entered are in range and all output should be clear with any floating point numbers displayed to two decimal places.

### Testing

You should carry out testing using the test cases given below. To do this, you should copy the list of test cases and the test table below into your word document that you will submit. You should then fill in the table with the data you would use to test the cases and your expected and actual results. Note your table will need additional rows. For the expected results, where it is a calculation, you should show below the table at least one calculation explicitly to show how you have done it. For example, you should enter all the expected moving mean values that you have calculated in the table but only include below the table, the details of the calculation for the January to March moving mean.

#### Test Cases

1. Entering numeric data above the range when entering rainfall and drought level
2. Entering numeric data below the range when entering rainfall and drought level
3. Entering numeric data that is in range when entering rainfall and drought level
4. Entering numeric data that is above the range when entering a choice
5. Entering numeric data that is below the range when entering a choice
6. Entering numeric data that is in range when entering a choice
7. Displaying the entered data works correctly
8. Displaying the sum of the values entered works correctly
9. Displaying the mean of the values entered works correctly
10. Displaying the standard deviation of the figures entered works correctly.
11. Displaying the median rainfall figure works correctly
12. Displaying the two lowest rainfall figures works correctly
13. Displaying the 3 month moving means works correctly
14. Displaying the drought months when there are no drought months works correctly
15. Displaying the drought months when all months are drought works correctly
16. Displaying the drought months when some months are drought works correctly
17. Displaying the near drought months when there are no near drought months works correctly
18. Displaying the near drought months when it is the same set of months as the drought months works correctly
19. Displaying the near drought months when there are additional months than just the set of drought months works correctly
20. Displaying the near drought months when all months are near drought months works correctly
21. The program ends correctly when option 10 is entered

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Data Used | Expected Result | Actual Result |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Assessment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Criterion | Fail | Pass | Good Pass | Excellent Pass |
| General Input, Output and Data Validation  *Menu, getValue, enterRain, enterDroughtLevel*  **3 Marks** | Data is not successfully obtained and/or not displayed clearly. | Data is successfully entered and displayed. | Data entry works, validation is okay and the output is clear. | Everything works well and good use is made of code reuse. |
| Initial data display and calculations  *displayData, calculateTotal, calculateMean, movingMean*  **3 Marks** | At least some parts do not work correctly. | All functions work. | all functions work well and are sensibly coded. | Everything works well and is efficient. Good use is made of code reuse. |
| Standard Deviation  *calculateStdDev*  **3 Marks** | Does not work correctly. | Works but not well coded. | Function works and the coding is sensible. | Function works with clear, efficient code. |
| Calculations relating to ordering  *calculateMedian, lowestTwoValues*  **3 Marks** | At least one does not work correctly. | Work out values correctly but not in an efficient way. | Work out values correctly and makes effective use of an additional function to do the pre-processing. | Works well both the functions asked for and the pre-processing are efficiently implemented. |
| Drought figures  *droughtMonths, nearDroughtMonths*  **2 Marks** | At least one does not work correctly. | Work but not in an efficient way. | Solve the problem effectively. | Solve the problem with neat, efficient code. |
| Main program  **3 Marks** | Does not work correctly. | Data is entered and options are carried out correctly | The program works okay, Input is all checked correctly and output is all formatted correctly. | Everything works well and is validated and formatted effectively. The code is neat and efficient. |
| Well-structured code  **2 Marks** | The code is not well structured. | The code is sensibly set out. | The code is well structured. | The code is very well structured and efficient in its operation. |
| Good use of python  **2 Marks** | Shows little understanding of some areas of python. | Shows a reasonable understanding of most concepts taught. | Shows a good understanding of all concepts taught. | Shows a good understanding of all concepts taught and demonstrates a high degree of skill in using them. |
| Testing  **4 Marks** | Testing is not attempted or incomplete. | Testing has been carried out successfully for all the test cases. | Testing has been carried out successfully with sensible and correct test data used throughout. | Testing has been carried out successfully with sensible and correct test data used throughout and all the calculations asked for are provided. |
| **Total 25 Marks** |  |  |  |  |

## Submission

As with assignment 1, you should submit a Word file containing a copy of your code and your testing and a .py file containing your code to Moodle.

1. The rainfall figures are of a fairly similar level to those for Great Britain but the drought level used is quite high. [↑](#footnote-ref-1)