

## **CITS1401 Project 1 grading guide, Semester 2, 2024**

There are 24 marks for functionality which are divided into 4 outputs test cases and a few error states. Each test case checks each output with several files. Marks for each test case are mentioned below.

Please provide feedback explaining the reasons for the student's loss of marks. Feel free to include any additional comments you find relevant. You can also include a note in the comments section using a format like  $x/24$  (test) +  $y/3$  (style) +  $z/3$  (efficiency) to further specify the breakdown of marks. Please note that you can't leave the comment section empty in Moodle. You can add their awarded marks for each criterion ( $x/24$  (test) +  $y/3$  (style) +  $z/3$  (efficiency)) in the comment section if they get full marks.

If the output is not returned in a proper format as required and simply printed, then deduct 4 extra marks as mentioned at the end and grade it accordingly. This is considered as marker's intervention to fix student's code.

Following table shows the marking criteria for each calculation they complete in Python:

Maximum awarded marks	Criteria
4	Maximum and minimum calculation.
6	Mean, median and mean absolute deviation calculation.
4	Standard deviation calculation.
6	Correlation calculation.
4	Error cases
6	Style and efficiency
Total: 30	

### **Output test cases:**

#### 1) Maximum and Minimum (4 Marks):

1 mark per correct ID in two test cases.

[max1, min1]

[max2, min2] = 4 marks

#### 2) Mean, Median and Mean Absolute Deviation (6 Marks):

1 mark per correct number in two test cases. 0.5 marks per correct but unrounded number.

[mean1, median1, deviation1]

[mean2, median2, deviation2] = 6 marks

#### 3) Standard Deviation (4 Marks):

2 marks for correct list of numbers in two test cases. 1.5 marks if numbers are correct but unrounded. 1.5 marks if list is not sorted. 1 mark if unrounded and unsorted.

#### 4) Correlation (6 Marks):

3 marks per correct number in two test cases. 2.5 marks per correct but unrounded number.

5) Error State (4 Marks):

For successfully passing each error state assign 1 mark. We are checking following error state:

- i) Zero division: text file containing only one line (product sales zero for both).
- ii) Case sensitive input
- iii) All data zero for a category
- iv) Single row for a category

Avoiding triggering a traceback and returning the outputs in the correct format are sufficient to allocate the mark.

**Style and efficiency: (be lenient but mention in comments to improve for next time)**

- Style (3/6) which involves intuitive variable and function names, consistent indentation, comments, etc.
  - Default is 3.
  - Deduct 0.5 mark if person's name or student id is not on the file to identify author of the code.
  - Deduct 1 marks if scant comments are provided.
  - Deduct 1 marks if no functions other than the main are created.
  - Deduct 0.5 mark if variable and function names are not intuitive or consistent indentation features are missing.
  - Minimum can be zero.
- Efficiency (3/6):
  - Default is 3 marks.
  - Deduct 1 mark if readline() function is used in a loop or file is opened multiple times.
  - Deduct 2 marks if the code includes repeated blocks instead of loops or code has more loops than necessary.
  - Deduct 1 mark if the code is taking long time to compile the big testing file.

For any intervention to fix the code, **deduct 4 marks extra**. However, it should be quick and simple. Normally missing a colon or wrong indentation or missing a character of some instruction is considered as a simple debug. Re-writing or adding or removing the code is not part of the grading and code should be graded accordingly as mentioned above.

**If student has imported any module, then comment it out and run the code, grade it accordingly and assign 0/3 in efficiency.**