Marking Sheet of Assignment 1 of BCPR301

Student Name/ ID ..........Zhiming Liu..........................................................................................

Other group members …………Vaishu Bhavu, Michael Barr……………………………………………………..

**You MUST supply (i.e., ZERO mark if not):**

1. A class diagram of your proposed program. And
2. A help file details the commands provided by your line-oriented command interpreter and the lecturer must approve these before you start the coding for this assessment. And
3. Your program must be able to do all the tasks mentioned in the section of Problem  
   domain. Please note that here displaying data does not mean simply outputting the  
   data as a 2D table. And
4. Your code MUST comply with the Python style (i.e., being able to pass PEP8 check).  
   And
5. A document to list (for each component claimed for marks in your program): a) the  
   ownership (i.e., done by you or someone else?); b) self-reflection on robustness1; and c) self-reflection on the completeness and implementation. And
6. You must carry out version control in a repository during your development process.  
   And
7. A filled self-marking sheet.

**Your Repository Link:**

https://github.com/heyerique/Python-Assignment1/tree/V001

**Marking guide (max 60 marks in total):**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Features | Used by your peers (2 mark) | Robustness (2 mark) | Complete and well implemented, i.e., “What is clever about this?” (2 mark) | Marks |
| 1 | Support command-line arguments |  |  |  |  |
| 2 | Has a line-oriented command interpreter based on cmd | 2 | 2 | 2 | 6 |
| 3 | Display command line help of available commands | 2 | 2 | 2 | 6 |
| 4 | Change commands and options | 2 | 2 | 2 | 6 |
| 5 | Validate data | 2 | 2 | 1 | 5 |
| 6 | Provides object-persistence / object serialization using either pickle or shelve |  | 2 | 2 | 4 |
| 7 | Can load data from a file | 2 | 2 | 2 | 6 |
| 8 | Can raise exceptions and provide exception handling | 2 | 2 | 2 | 6 |
| 9 | Amount of error trapping & handling | 2 | 2 | 2 | 6 |
| 10 | Provide doctests | 2 |  | 2 | 4 |
| 11 | Provide unittests | 2 |  | 2 | 4 |
| 12 | Breadth of test coverage | 1 |  | 1 | 2 |
| 13 | Can deal with directories and file locations | 2 |  | 2 | 4 |
| 14 | Pretty print, i.e., displaying data in bar chart, pie chart, etc. | 2 |  | 2 | 4 |
| 15 | Can save and read data from a database |  |  |  |  |
| Total |  | 23 | 16 | 24 | 63 |

|  |
| --- |
| **Marks** |
|  | 0 | 1 | 2 |
| **Used by peers** | Not used by any peer | Half of the team members use | All team members use |
| **Complete and well implemented except doctest, unittest and test coverage features** | Not complete | Complete, but not very Pythonic | Complete and very Pythonic |
| **Complete and well implemented for doctests** | No doctest | < 5 different doctests | >= 5 different doctests |
| **Complete and well implemented for unittests** | No unittest | < 5 different unittests | >= 5 different unittests |
| **Complete and well implemented for breadth of test coverage** | < 18 different doctests and/or unittests | >= 18 different doctests and/or unittests | >= 36 different doctests and/or unittests |
| **Robustness** | Not be able to run during demonstration | Encounter some exceptions during demonstration | Encounter ZERO exception during demonstration |