Self-Marking Sheet & Reflection

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # No.  (marks a + b + c) | Component | a). Ownership | b). Robustness | c). Completion | |
| 1  (2 + 2 + 2) | Support command-line arguments | Brandon | The system handles all commands in 2 ways.  1. As a shell command  2. As a command line argument  When required arguments are not provided to a command there is a helpful warning printed  Extra arguments are ignored at this time  From the command line there is no way to execute more than 1 command  Commands can be easily added with NO need for more ‘if-else’ nesting and NO need for more switch cases to be added | Working. Some user guidance features missing. No exceptions are encountered  Extra arguments should present a warning  Should be a way to execute multiple commands from the command line as one command  The help command can be used in the shell and as a command line argument. The command displays a list of commands and a brief description. There is no way to filter help to a specific command | |
| 2  (2 + 2 + 2) | Has a line-oriented command interpreter based on cmd | Brandon |
| 3  (2 + 2 + 1.5) | Display command line help of available commands | Brandon |
| 4  (2 + 2 + 2) | Change commands and options | Alex | Added some commands | | |
| 5  (2 + 2 + 1) | Extract data | Brandon | The following items are extracted from python source code:   * Class names * Methods * Object Attributes * Class Attributes * Privacy based on naming convention starting with ‘\_’ or ‘\_\_’ | | Relationships are not detected  Only the first method argument is detected |
| 6  (2 + 2 + 1.5) | Validate data | Alex | The validation method allows checking and validation of a file to make sure that it is the right format. | | This is complete, the only thing I could add on to it is at the end of the command closing the file as it is left open which uses the machines resources. |
| 8  (2 + 2 + 1.5) | Can load data from a file | Alex | The code allows users to input a file using an input command; they are also able to put multiple files at once.  It also handles loading files from directories. | | I believe this code is complete, the only thing that could be added is being able to load multiple files from multiple directories. |
| 9  (2 + 2 + 1.5) | Can raise exceptions and provide exception handling | Alex | In the validation of the file I have added the most common exceptions to output an error message to allow the user to know what the error that they have incurred is. | | There is always a list of more exceptions that could be added but I decided to add the most likely ones. |
| 10 (2 + 2 + 2) | Amount of error trapping & handling | Brandon  Alex | All unhandled exceptions are prevented by checks and messages are displayed | | Extra command arguments should present a warning |
| 11  (2 + 2 + 1.5) | Provide doctests | Brandon – 2 tests  Alex – 3 tests | 2 Tests were made to test the significant parts of my implementation | | Does not cover many aspects of the code |
| 12 (2 + 2 + 1) | Provide unittests | Brandon – 3 tests  Alex – 3 tests | 3 Tests were made to test the significant parts of my implementation | | Does not cover many aspects of the code |
| 14  (2 + 2 + 1.5) | Can deal with directories and file locations | Alex | With the input I have added that the input can deal with different file directories/locations | | The only thing to add to it’s completeness is adding that you can do multiple files from multiple locations. |
| 15  (2 + 2 + 1) | Pretty print, i.e., displaying data in bar chart, pie chart, UML diagram | Brandon | See 5. Extract data | | |

**Expected marks for Brandon ~39/60**

**Expected marks for Alex ~44.5/60**

# Code Quality

The has decent quality. Validates under PEP8. Good code reuse.

See code base for various examples throughout. The CLI class has great examples of the before mentioned especially good code reuse, maintainability and expandability.

Git Repository

<https://github.com/BrandonDR/PR301-UML-Diagram>

Design Level UML Class Diagram

