Mark Schedule of Assignment 1

Student Name/ID

**You must supply (i.e., ZERO mark for refusing):**

1. A class diagram of your proposed problem domain and
2. A help file details for your line-oriented command interpreter and these must be approved by the lecturer before you start the coding for this assessment. And
3. Your code obeying the Python style (i.e., be able to pass PEP8 check). And
4. A document to list (for each component claimed for marks in your program) a) the ownership (i.e., done by you or someone else?); self-reflection on adequate quality; and c) self-reflection on the completeness and implementation.

**Marking schedule (max 10 marks):**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Components | Used by another one, i.e., Marks \*= 0.5 | Only used by yourself, i.e., marks \*= 0 | Adequate quality, i.e., “How is this robust?” (0.5 mark) | Complete and well implemented, i.e., “What is clever about this?” (0.5 mark) | Marks |
| load data from a file |  |  |  |  |  |
| Supports command-line arguments |  |  |  |  |  |
| display command line help of available commands |  |  |  |  |  |
| Has a line-oriented command interpreter based on cmd |  |  |  |  |  |
| change options |  |  |  |  |  |
| validate your selections |  |  |  |  |  |
| pretty-print, i.e., for a turn sequence |  |  |  |  |  |
| Provides object-persistence / object serialization using either pickle or shelve, i.e., saving (serializing) a turn sequence |  |  |  |  |  |
| Raises exceptions and provides exception handling |  |  |  |  |  |
| Amount of error trapping & handling |  |  |  |  |  |
| Provide doctests |  |  |  |  |  |
| Provide unittests |  |  |  |  |  |
| Breadth of test coverage |  |  |  |  |  |
| Can deal with directories and file locations |  |  |  |  |  |
| Total |  |  |  |  |  |