

Marking Sheet of Assignment 1 of BCPR301

Student Name/ ID

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

- 1 A class diagram of your proposed program. 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 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:

Marking guide (max 60 marks in total):

	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

¹ **Robustness.** The degree to which a system continues to function in the presence of invalid inputs or stressful environmental conditions.

Complete and well implemented for doctests	No doctest	< 3 different doctests	>= 3 different doctests
Complete and well implemented for unittests	No unittest	< 3 different unittests	>= 3 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

	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				
3	Display command line help of available commands				
4	Change commands and options				
5	Validate data				
6	Provides object-persistence / object serialization using either pickle or shelve				
7	Can load data from a file				
8	Can raise exceptions and provides exception handling				
9	Apply error trapping & handling				
10	Provide doctests				
11	Provide unittests				

12	Breadth of test coverage				
13	Can deal with directories and file locations				
14	Pretty print, i.e., displaying data in bar chart, pie chart, etc.				
15	Can save and read data from a database				
	Total				