

BCDE321 Advanced Programming

Assessment Two

Practical Assessment 2

Semester One, 2020

Due date: Monday, 23 March 2020

Time: 5pm

Student Name/ ID

Ara and its division members reserve the right to use electronic means to detect and help prevent plagiarism. Students agree that when submitting this assessment, it may be subject to submission for textual similarity review to Turnitin.com.

Submissions received late will be subject to a penalty of 10% of the student's mark per working day.

This assessment is worth 25% of the total marks for this course.
This paper has three (3) pages including the cover sheet.



This assessment relates to the following Learning Outcome(s):

- Demonstrate ability to integrate software components and frameworks

Problem Domain

Create a Python 3 program, which can generate a UML 2 diagram based on the source code of a Python 3 program in a file/ files under a directory by using Python package(s).

Note that real-world data is often not perfect. It means that you should check/ wash the input data before storing and displaying them.

Requirements for Submission

Every student **MUST** submit the followings as a single .zip file into the dropbox on course Moodle site by the deadline indicated; otherwise, **ZERO** mark will be given.

- 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 coding 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 an online repository during your development process. The URL link of your online repository needs to be provided in your self-marking sheet. And
- 7 A filled self-marking sheet.

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

Requirements in Detail

- The quality goal is to create a ROBUST system.
- You MUST work with your classmates for this assessment, as pair programming teams. You should clearly identify the component you worked on. Only your OWN work will be awarded marks. Marks will also depend on if your code exists in the final program that your team submits at the end.
- You can select which component you work on from the list provided below.
 - 1) Support command-line arguments
 - 2) Has a line-oriented command interpreter based on cmd or similar package
 - 3) Display command line help of available commands
 - 4) Change commands and options
 - 5) Extract data
 - 6) Validate data
 - 7) Provides object-persistence/ object serialization using either pickle or shelve
 - 8) Can load data from a file
 - 9) Can deal with file directory
 - 10) Can raise exceptions and provide exception handling
 - 11) Amount of checking for pre- and post- conditions of methods
 - 12) Provide doctests
 - 13) Provide unittests
 - 14) Pretty print, i.e., displaying data in chart/ diagram, e.g., bar chart, pie chart, UML diagram, etc.
 - 15) Can save and read data from a database, e.g., SQLite, MySQL and MongoDB
- You will be marked on a 0 – 6 scale for each component for which you present original source code. Your total marks will be the sum of the marks for each component. This assessment will be marked out of 60.
- When marking the assessment, you will be asked to explain in front of the class or tutor how your code works.