BCPR301 – Advanced Programming
Assessment4

This assessment is worth **10% of the marks** out of the total of 100 for the whole course grade for BCPR301.

Deadline

- You must hand in all required materials by 10am, Wednesday, 1 May

This assessment relates to the following learning outcomes:
2. Learn and apply a new programming language.
3. Appropriately identify and apply techniques for integrating software components and frameworks.
4. Evaluate the effectiveness of work, and make recommendations accordingly.

This is an INDIVIDUAL assignment. Each student must work on their own DIFFERENT program. You will be responsible for obtaining and installing the framework and all other necessary resources for you to complete this assignment.

Tasks:

To analyze and document bad smells in the programs provided (i.e., one of Assignment 1 solutions provided), and then refactor the programs using current "best practice" programming techniques. Please do your best to use design principles[1] and ideas of Object-Oriented Programming (OOP) in your discussion and evaluation in this assignment.

You must also submit your **self-marking sheet** to indicate how many marks you think you can get for each part based on the marking guide provided below. **No marks** will be given if you do not submit your self-marking sheet.

After you complete your assignment, please put the URL link of your remote repository and self-marking sheet together in a MS Word file and only submit that file. By following the link that you provide, all other files including source code required by this assignment should be able to be retrieved.

It is a good idea to always refer to the line number and file name (plus file path) in order to clearly state which piece of the code you are talking about in your answers.

Marking guide:

There are 13 * N marks in total in this assignment where N = 3.

1. Smell detection (4 * N marks)

    1) Identification of N bad smells in the programs provided. For the sake of learning, please try to identify **different types of bad smells**. (N marks)

[1] https://www.codeproject.com/Articles/567768/Object-Oriented-Design-Principles

2) The location of each bad smell identified (N marks)
3) And discussion on the reasons why you think that the ones you identify are bad smells in a concise fashion. Please do not simply copy general reasons from somewhere and paste them in your submission (N marks)
4) And brief discussion on the refactoring strategies/ approaches you are going to use to remove each bad smell (N marks)

Please document the bad smells in the following format:
Name: <e.g., duplication>
Location: <packageName/folderName>-<moduleName/fileName>-<className>-<methodName>-<between Line XXX and XXX>
Reasons:
1. …
2. …
…
Strategies/ approaches: <e.g., extract method>

Note: marks are for the number of bad smells correctly identified and sound reasons.

2. Tests development (4 * N marks)

1) To develop a set of tests for the methods/ classes/ modules/ packages encompassed by the bad smells you previously identified (3 * N marks)
2) Please also use coverage package to generate a HTML report in order to show your code branch coverage **== 100%**. And all tests should be able to be run together by running a single .py file (N marks)

Note: your testing code needs to pass PEP8 check as well.

3. Refactoring (5 * N marks)

In order to remove the bad smells that you previously identified, you need to follow the refactoring process that we discussed in class sessions.

1) Identifying the worst smell and the reasons why it is the worst one (N marks)
2) Version control via a remote repository and testing (N marks)
3) Modification to remove the worst smell and PEP8 validation (2 * N marks)
4) Effectively evaluations (N marks)

Note: removing a single bad smell often requires a number of refactoring cycles. During a cycle, a small modification should be implemented. After each implementation, you need to make a "commit" to your remove repository, e.g., GitHub, for version control.

In your repository, it is better that you have a document to briefly step by step explain your refactoring process for each individual bad smell identified and a discussion on how well you remove the bad smells (e.g., has the bad smell successfully been removed at the end? Did you bring new bad smells into the program? How well is your program now in terms of software quality?)

Your source code needs to pass PEP8 check.