

## 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 **Wednesday, 4 May 2016, 10 AM.**

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 analyse and document bad smells in the programs provided, and then refactor the programs using current “best practice” programming techniques. Please do your best to use design principles and ideas of 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 marking part based on the marking guide provided below. **No marks** will be given if you refuse to submit your self-marking sheet.

After you complete your assignment, please put the URL link of your remote repository in a .txt file and only submit that file. By following the link you provide, all the files required by this assignment should be able to be retrieved.

### Marking guide:

**There are 12 \* N marks in total in this assignment where N = 3.**

1. Smell detection (4 \* N marks)

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)

And discussion the reasons why you think that the ones you identify are bad smells in a concise fashion (N marks)

And refactoring strategy/ approach 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>-  
<between Line XXX and XXX>

Reasons:

1. ...
2. ...
- ...

*Strategy/ approach:*     <e.g., extract method>

*Note: Marks are for the number of bad smells correctly identified and sound reasons*

## 2. Tests development (4 \* N marks)

To develop a set of tests for the methods encompassed by the bad smells you previously identified (3 \* N marks)

Please also use coverage package to generate reports in order to show your code coverage  $\geq 80\%$ . 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.*

## 3. Refactoring (4 \* N marks)

To refactor the methods encompassed by the bad smells you previously identified in order to remove the bad smells, you need to follow the refactoring process we discussed in class sessions.

Version control via a remote repository and testing (N marks)

Modification and PEP8 validation (2 \* N marks)

Effectively evaluations (N marks)

Note: removing a single bad smell often requires a number of refactoring cycle. During a cycle, one a small modification should be implemented. After each implementation, you need to make a “commit” to your remote repository, e.g., GitHub, for version control. In your repository, you need to 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 smells successfully been removed at the end? How well it is? Do you bring new bad smells into the program?)

*Note: Your source code needs to pass PEP8 check.*