**BCPR301 – Advanced Programming**

**Assessment 2 Marking Sheet for Coder**

Student Name Zhao(Jerry) Wang

# The compulsory (i.e., ZERO mark if not being provided):

1. You MUST supply a filled self-marking sheet to indicate how many marks you think you can get for each part based on the marking guide provided below.
2. A feature list and an interface diagram of the target source code

# Your repository link:

<https://github.com/jerry8812/PR301Repo#pr301repo>

# Marking guide (max 9 \* N marks in total where N = 3):

1. Smell detection (4 \* N marks)
   1. Identification of N+ bad smells in the target source code. For the sake of learning, you need to identify **bad smells from different bad smell categories (excluding comments bad smell)** covered in our class (N marks)
2. Duplicated code
3. Middle man bad smell
4. Lazy class
5. Marks: 4 \* 3 = 12
   1. The location of each bad smell identified (N marks)
6. Duplicated code bad smell

location: <https://github.com/jerry8812/PR301Repo/tree/master/Assginment1>

TIGrEx.py line 128-133,141-146,154,159,168-175,184-189,198-205,214-221,230-236,244-250,264-271,291-296

1. Middle man bad smell

location: <https://github.com/jerry8812/PR301Repo/tree/master/Assginment1>

TIGrExParser.py line 69-91,93-97

1. Lazy class bad smell
2. location: <https://github.com/jerry8812/PR301Repo/tree/master/Assginment1>

TIGrExSourceReader.py

The entire class

1. Marks: 3
   1. The reasons why you think that the ones you identify are bad smells in a concise fashion (N marks)
2. Duplicated code bad smell: This class TIGrEx.py which had the same checking code duplicated

Many times into several methods

1. Middle man bad smell: This class TIGrExParser.py had several methods which do nothing but navigate to Drawer
2. Lazy class bad smell: Source Reader "TIGrExSourceReader.py" Class only had one method which just pipe self.source to itself object self.parser.
3. Marks: 3
   1. Brief description about the refactoring strategies/ approaches you are going to use to remove each bad smell (N marks)
4. Duplicated code bad smell: I’m going to extract all the error checking code to one method and call this method before do any command. first, I created a method named Error\_Check and implemented check in this method. Secondly, updated one do\_method, test this method. finally, update all do\_method. Test all of them again with enter wrong command on purpose such as ‘west’, ’fawefaw’, ’west 100aafw’. But it turns out to be another duplicated code as all the Methods call error check method and I am going to fix it in second bad smell
5. Middle man bad smell: There are several methods that are actually doing nothing but navigating to Drawer. What I am going to do is delete all the middle man methods, redirected them in method Parse. Regarding of the first bad smell, I am going to put the data check in parse and call method parse in do\_methods in TIGrEx.py to resolve duplicated code. After refactoring, all do\_method in TIGrEx.py get command from cmdloop, convert command and data to list then pipe them to Parser.parse to do error checking then call drawer to implement command. Test with all the command again.
6. Lazy class bad smell: Get rid of entire Source Reader "TIGrExSourceReader.py" Class, add Attribute self.script\_extenion at class TIGrEx, call pase to run script straightaway. Test with all the command again.
7. Marks: 3
8. 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)

I considered the Middle man bad smell is the worst one. It impacted the whole structure of this project. The Parser class meant to parse all the sources data and then call Drawer to draw instead of doing no perform. In additional, doing data check in parse method can resolve duplication in class TIGrEx.

Marks: 3

1. Version control via an online repository (N marks)

As showed on Github

Marks: 3

1. Modification to remove the worst smell and PEP8 validation (2 \* N marks)

Modification and PEP8 check done

Marks: 2\* 3 = 6

1. Testing and effectively evaluations on your refactored code in a concise fashion (N marks)

All doctests passed