## BCPR301 Assignment 2

## Smell Detection 1 Marks -4

Name: <Large Class>

Location: <Assignment2>-<employee\_database.py>-<EmployeeDatabase>- <get\_all\_employee(),get\_average\_salary(),get\_ave\_sales()>-<between Line 63 and 125>

Reasons:

* EmployeeDatabase class contains lots of method that get data for use of other classes.
* It makes the class bigger and basically database connection class should be separated from other classes for security reasons and call whenever necessary.
* This class may grow bigger when building this program further therefore it is useful to separate get data methods into another class

Strategies/ approaches: <extract method>

Create a new class called GetData and add the above methods (<get\_all\_employee(),get\_average\_salary(),get\_ave\_sales()>) inside that class. Necessary changes were made in other classes (Controller). Program works properly after this editing.

## Smell Detection 2 Marks -4

Name: <Switch Statement>

Location: <Assignment2>-<command.py>-<Command>- <do\_chart()>-<between Line 86 and 110>

Reasons:

* Though it is not a bad syntax “if conditions” comes under switch statement bad smell
* This coding (if conditions) can be under code smell
* It may violate open close principle in OOP – when user need to add new codes (conditions) he needs to modify existing code again and again
* It is difficult to maintain because it is inside a single method when adding new code method will grow and maintenance will be hard
* This will lead to code redundant

Strategies/ approaches: <replace conditional with polymorphism>

Do\_chart() method has different actions such as print sales chart, print pie chart and print line chart according to the options given. It is possible to create sub classes for all conditions and create method inside those classes to reduce the bad smell. Change all other files that use this method.

## Smell Detection 3 Marks -4

Name: <Long Method>

Location: <Assignment2>-<command.py>-<Command>- <do\_load()>-<between Line 18 and 56>

Reasons:

* do\_load() method contain lots of lines (38)
* it is difficult to read code and it smells bad
* it is possible to separate the coding using comments

Strategies/ approaches: <extract method>

Give comments to identified parts in the codes

Create new methods for each comments and load those methods inside the do\_load() method

## Smell Detection 4 Marks -4

Name: <Divergent Change>

Location: <Assignment2>-<command.py>-<Command>- <greeting(), set\_name() and set\_number\_of\_command() >-<between Line 123 and 155>

Reasons:

* Command class is mostly about line oriented commands but it includes Command – line arguments such as greeting() and set\_number\_of\_command()
* If they are included in another class it will be easy to change program easily when needed.

Strategies/ approaches: <extract method>

Created a new class called CommandLine and added the methods inside that class. Changed Command class as necessary.

**Sub Total - 16**

## Tests development (4 \* N marks)

1. Test - Marks 2\*4
2. Coverage - Marks 3

**Sub Total - 11**

## Refactoring

1. Identifying the worst smell and the reasons why it is the worst one (2 marks)

Worst smell is **2. <Switch Statement>** because if this bad smell affect most part of the program. And if it is removed the program will be easy to maintain and understand.

2) Version control via a remote repository and testing (4 marks)

3) Modification to remove the worst smell and PEP8 validation (2 \* 4 marks)

4) Effectively evaluations (2 marks)

**Sub Total – 16**

**Total - 43/52**