# Comment Smell

**Name:** Comment Smell

**Location:**

*FOLDER\_NAME* = badsmell\_assignment2\_python

*FILE\_NAME* = reader.py

*CLASS\_NAME* = FileReader

*METHOD\_NAME* = Read()

*BETWEEN\_LINES* = 5 and 27

**Reasons:**

1. This could be also considered a long method, but it only makes sense on what is happening because of the comments
2. If I extracted the different methods then they could become useful to use again separately rather than calling it as one method
3. If this method needed to be extended it would be quite difficult in its current state

**Strategies/ approaches:** Extract Method:

Extract the methods that are under each of the comments. The name of the methods would be related to what the comment said.

# Switch Statement Smells

**Name:** Switch Statement Smell

**Location:**

*FOLDER\_NAME* = badsmell\_assignment2\_python

*FILE\_NAME* = controller.py

*CLASS\_NAME* = Controller

*METHOD\_NAME* = valid()

*BETWEEN\_LINES* = 25 and 43

**Reasons:**

1. The code looks confusing from the massive list of if statements that call different methods
2. Making this cleaner would improve readability
3. If more functionality needed to be added then the if / else would just get longer

**Strategies/ approaches:** Extract Method, Polymorphism

Some of the methods may need to be extracted so that the methods can be run through one method call. The different method calls will be stored in a dictionary with the flag as the key and the method as the value. This will replace the if / else statements making the code more readable and flexible.

**Name:** Switch Statement Smell

**Location:**

*FOLDER\_NAME* = badsmell\_assignment2\_python

*FILE\_NAME* = commandLine.py

*CLASS\_NAME* = DomainCMD

*METHOD\_NAME* = arg()

*BETWEEN\_LINES* = 21 and 33

**Reasons:**

1. The code looks confusing from the massive list of if statements that call different methods
2. Making this cleaner would improve readability
3. If more functionality needed to be added then the if / else would just get longer

**Strategies/ approaches:** Extract Method, Polymorphism

Some of the methods may need to be extracted so that the methods can be run through one method call. The different method calls will be stored in a dictionary with the flag as the key and the method as the value. This will replace the if / else statements making the code more readable and flexible.

**Name:** Switch Statement Smell

**Location:**

*FOLDER\_NAME* = badsmell\_assignment2\_python

*FILE\_NAME* = controller.py

*CLASS\_NAME* = Controller

*METHOD\_NAME* = db\_table()

*BETWEEN\_LINES* = 58 and 86

**Reasons:**

1. The code looks confusing from the massive list of if statements that call different methods
2. Making this cleaner would improve readability
3. If more functionality needed to be added then the if / else would just get longer

**Strategies/ approaches:** Extract Method, Polymorphism

Some of the methods may need to be extracted so that the methods can be run through one method call. The different method calls will be stored in a dictionary with the flag as the key and the method as the value. This will replace the if / else statements making the code more readable and flexible.

**Name:** Switch Statement Smell

**Location:**

*FOLDER\_NAME* = badsmell\_assignment2\_python

*FILE\_NAME* = controller.py

*CLASS\_NAME* = Controller

*METHOD\_NAME* = pygal()

*BETWEEN\_LINES* = 118 and 132

**Reasons:**

1. The code looks confusing from the massive list of if statements that call different methods
2. Making this cleaner would improve readability
3. If more functionality needed to be added then the if / else would just get longer

**Strategies/ approaches:** Extract Method, Polymorphism

Some of the methods may need to be extracted so that the methods can be run through one method call. The different method calls will be stored in a dictionary with the flag as the key and the method as the value. This will replace the if / else statements making the code more readable and flexible.

**Name:** Switch Statement Smell

**Location:**

*FOLDER\_NAME* = badsmell\_assignment2\_python

*FILE\_NAME* = controller.py

*CLASS\_NAME* = Controller

*METHOD\_NAME* = pickled()

*BETWEEN\_LINES* = 146 and 164

**Reasons:**

1. The code looks confusing from the massive list of if statements that call different methods
2. Making this cleaner would improve readability
3. If more functionality needed to be added then the if / else would just get longer

**Strategies/ approaches:** Extract Method, Polymorphism

Some of the methods may need to be extracted so that the methods can be run through one method call. The different method calls will be stored in a dictionary with the flag as the key and the method as the value. This will replace the if / else statements making the code more readable and flexible.

# Duplicated Code Smells

**Name:** Duplicated Code Smell

**Location:**

*FOLDER\_NAME* = badsmell\_assignment2\_python

*FILE\_NAME* = controller.py

*CLASS\_NAME* = Controller

*METHOD\_NAME* = pickled()

*BETWEEN\_LINES* = 141 and 151

**Reasons:**

1. If I wanted to add more functionality I’d have to copy paste the code again to add the new functionality which is bad practice
2. Might be able to add new functionality
3. The code doesn’t need to be duplicated

**Strategies/ approaches:** Extract Method, Pull Up Method

Pull up the method so that it only needs to be called once and use a dictionary for the method selection

**Name:** Duplicated Code Smell

**Location:**

*FOLDER\_NAME* = badsmell\_assignment2\_python

*FILE\_NAME* = display.py

*CLASS\_NAME* = PyGal

*METHOD\_NAME* = bar\_char()

*BETWEEN\_LINES* = 8 and 22

*FILE\_NAME* = controller.py

*CLASS\_NAME* = Controller

*METHOD\_NAME* = pygal() & py\_view()

*BETWEEN\_LINES* = 111 and 137

**Reasons:**

1. If I wanted to add more functionality I’d have to copy paste the code again to add the new functionality which is bad practice
2. Might be able to add new functionality
3. The code doesn’t need to be duplicated

**Strategies/ approaches:** Extract Method, Pull Up Method, Polymorphism

Need to fix the if statement in the controller and then maybe use a dictionary to hold the data for the bar chart. This would mean only one input parameter and allow it to grow or shrink to fit what is needed. This would remove the duplication

**Name:** Duplicated Code Smell

**Location:**

*FOLDER\_NAME* = badsmell\_assignment2\_python

*FILE\_NAME* = database.py

*CLASS\_NAME* = DatabaseMaker

*METHOD\_NAME* = bar\_get()

*BETWEEN\_LINES* = 81 and 93

**Reasons:**

1. Is the same as get()
2. If I wanted to add more functionality I’d have to copy paste the code again to add the new functionality which is bad practice
3. Might be able to add new functionality
4. The code doesn’t need to be duplicated

**Strategies/ approaches:** Extract Method, Pull Up Method

Pull up the method so that it’s the same as the other method with only one small change or create an abstract class that handled the difference between the two pieces of code.

# Divergent Change Smell

**Name:** Divergent Change Smell

**Location:**

*FOLDER\_NAME* = badsmell\_assignment2\_python

*FILE\_NAME* =controller.py

*CLASS\_NAME* = Controller

*METHOD\_NAME* = pickled\_write() / pickled\_read()

*BETWEEN\_LINES* = 167 and 178

**Reasons:**

1. This should be its own class
2. The controller should only be calling the method not doing it

**Strategies/ approaches:** Extract Class

Move the pickled implementation to its own class so that the controller only references it not makes it.