# Bad Smells

|  |  |
| --- | --- |
| Name: | Primitive Obsession |
| Location: | src/controller – controller.py – Controller – chart\_bar and chart\_pie  – Lines 120, 121, 127, 128 |
| Reasons: | 1. The primitives are storing the indexes of lists passed to the graph package. 2. These primitives are explaining a part of the code that can be rewritten in a much easier to understand way. |
| Strategies/ approaches: | 1. Create a class to fix and store the list. 2. Give the class methods to get the label and data parts of the list, meaning no primitives being used for list parts. |

|  |  |
| --- | --- |
| Name: | Dead Code |
| Location: | src/logger |
| Reasons: | 1. Whole package was used for debugging of data file reading. 2. Package is not used in assignment 1 hand in. 3. Creates extra project size and baggage. |
| Strategies/ approaches: | 1. Remove the whole package to a folder outside the project. 2. Removal of the package from the project repository for further updates. |

|  |  |
| --- | --- |
| Name: | Duplicate Code |
| Location: | src/validator  *Parent class*: validate\_field.py – ValidateField – get\_field – Line 20  Child Classes:   * bmi.py – BMI – get\_field – Line 21 * emp\_id.py – EmpID – get\_field – Line 21 * gender.py – Gender – get\_field – Line 21 |
| Reasons: | 1. Adds complexity. 2. Looks ugly and stupid. 3. Occurred from programmer unfamiliarity with the language. |
| Strategies/ approaches: | 1. Pull up method: Moving the method to the parent and overriding the method when a string is not the return value. |

|  |  |
| --- | --- |
| Name: | Duplicate Code |
| Location: | src/validator – dob.py – DOB – get\_db\_friendly and get\_field – Lines 59 to 63 |
| Reasons: | 1. The get\_field method only calls the get\_db\_friendly method and nothing else. 2. This adds unnecessary bloat and complexity to the class, because get\_db\_friendly is completely unnecessary. |
| Strategies/ approaches: | 1. Transfer the internals of get\_db\_friendly to get\_field. 2. Remove get\_db\_friendly. |

# Worst Smell

Smell 3 is the worst smell because it adds the most unnecessary lines of code into the program. Smell 2 cannot be the worst as this whole package can be completely ignored in future development as it is not currently used in the program.

# Evaluation

## Bad Smell 3: Duplicate Code:

* Move the get\_field method to the parent class and remove the method from EmpID, BMI, Gender
* Create a class ValidateNumberField as a child of ValidateField and parent of Sales, Salary, Age. Move the get\_field method and the declaration for self.\_field\_number = 0 from the children to the parent.

Duplicate code is removed, class structure is much improved. No new smells introduced.

## Bad Smell 2: Dead Code:

* Delete the package
* Remove all references to the unused class from controller

Removal of the dead code reduces the baggage, improving the program without introducing new smells.

## Bad Smell 1: Primitive Obsession:

* Create a ABC and class DataFixer that:
  + Takes the list from db query
  + Separates the list into label and data portions
  + Has get methods for label and data lists
* Remove the fix\_list method from Controller
* Use a DataFixer object to access the data for graphing

Code is now far easier to read in the Controller chart\_pie and chart\_bar methods.

Primitives have been removed and need for comments no longer exists.

No new smells introduced.

## Bad Smell 4: Duplicate Code:

* Transfer the content of get\_db\_friendly to get\_field
* Delete get\_db\_friendly

Removal of the method removes the smell without creating any other problems, removing unnecessary bulk from the code.