1. Explain how object oriented design can ensure lesser code smells and easier maintenance.

The use of object oriented design convinces developers to make use of the fundamental object oriented programming (OOP) principles such as encapsulation, abstraction, inheritance, and polymorphism. It reduces the chances of code coupling with other unrelated parts of code and makes them cohesive by serving only 1 specific purpose. It makes it easier to maintain because developers can tell which part of the code to look for at a glance when a bug occurs for a particular class, and code smells that are not object oriented design related are reduced because closely related members in a class are separated from normal code to prevent side effects.

1. Explain three benefits of refactoring. After that, list of five refactoring opportunities or Code smell.

**3 benefits of refactoring**

* Increases the maintainability of code. Developers will be able to make changes to the code even after a long hiatus of developing the code.
* Increases understandability of code. Developers can easily understand what the functionality of a particular code is at a glance.
* Better code organization and structure. Code organization is more tidy and neat which allows developers to nagivate more easily around the codebase.

**5 Code smells**

* Bloaters
* Object oriented abusers
* Change preventers
* Dispensibles
* Couplers

1. Explain the aspect of 1) reason for the code smell problem, 2) treatment, and 3) advantage gained from code refactoring of the code smell area below:

i) Refused Bequest

1) reason for the code smell problem

A subclass attempts to inherit from a parent class for the purpose of reusing the parent’s members, but subclass and superclass are completely different, and the other unneeded methods go unused or redefined to give off exceptions.

2) treatment

Delegate the required members of a parent class and eliminate inheritance.

3) advantage gained from code refactoring of the code smell area

Reduces chances of class properties and methods giving exceptions and lesser memory used.

ii) Shotgun surgery

1) reason for the code smell problem

A single responsibility is split up among a large number of classes

2) treatment

Move methods related to the responsibility to a common class

3) advantage gained from code refactoring of the code smell area

Less duplication, easier maintenance, better organization