Joe Chanda 3/7/2016

COSC 603 – Project 2

**Task 6**

Renaming is a refactoring feature that provides an easy way to rename identifiers for code. The renaming of the class setTheOwner() was a mistake on Eclipse’s refactoring. It changed another parameter that we didn’t want changed. The rename refactor is not like find/replace. The word owner was not change everywhere, just in that class and other parameters calling it.

**Task 7**

This hierarchy techniques are useful when dealing with variables of similar. It showed the difference between setAvailable and isAvavbilable.

**Task 8**

From how I see it, extracting interfaces use the same subset of a class's interface, or two classes have part of their interfaces in common. I extracted iOwnable for the available, the name, and the owner subclasses

**Task 9**

Ectracting a method allows you to move code to a separate new method in order to replace the old code with a call to the method. I chose the getRent method since it could be the most easily used with others to call it.

**Task 10**

Creating a local variable is a nice way of keeping things simple you can have a declaration mean a paragraph or two of code and just point to that. This can save a lot of development time and confusion later when understanding extra non-necessary code.

**Task 11**

I learned you can use the change method signature refactoring to change parameter name and types as well as change the method name and return type.

**Task 15**

I learned what refactoring is and why it’s important. Refactoring is when you edit code, but not changing what the program is doing. There are many different methods to refactoring. All of which have the same goal, that is to make code smell better. It is important to constantly preform unit tests since you can unintentionally create errors.

The main reason for refactoring is for the next guy. Most likely you will not be developing your own code then going back to try to understand it. You need to refactor to make your own work easier to maintain for the next person.

Eclipse has many different refactoring tools to use. The nice thing about them is they are pretty strait forward and not too complicated. The tools are so easy to use that anyone can figure them out. The Eclipse refactoring wizard makes the job easier by automatically detecting potential issues and modifying the code.

JDeodorant is an Eclipse plug-in that identifies design problems in software, known as bad smells, and resolves them by applying appropriate refactoring. JDeodorant is extremely helpful. I think of it like a spell checker. It shows me everything that smells bad and shows me a way to fix it.

Unit testing is important because it allows you to update code without fear. Everything you do can always be tested. It’s like experimenting with a safety net. You can test your code right away to make sure it still works. Unit tests mean that you can make changes to your code and instantly know whether those changes have broken it. When you have a good baseline of tests, you know your code works and you don’t have to be afraid of changes to it in the future.