COSC 603

Software Maintenance & Testing

Spring 2016

**Project #2 – Refactoring and Design Smells**

Name : Marlene Encinas

**Task 7 – Refactoring: Changing a Class Hierarchy.**

***For this task, briefly describe your experience with this task and for which design smells pushing down or pulling up a class’s field(s) and/or associated methods could help make the code more maintainable and why.***

Answer:

Following the steps describe in the Task 7 was straight forward. Seeing Junit in action reporting failures when a change generates undesirable behaviors demonstrate how Eclipse can be a useful time saving tool at the time of changing a feature or reverting the change. The impacts of doing these manually would be catastrophic and tedious doing manually.

Push up and Push down can be applied when:

- a class uses methods of another class excessively.

- a class has dependencies or implementation details of another class.

- a class that overrides a method of a base class in such a way that the contract of the base class is not honored by the derived class.

- a class that does too little

Push up can be used when same code exists in two unrelated classes and probably they need to be related.

Using Eclipse’s Rename refactoring will make those changes throughout the entire project intelligently, without user intervention because Eclipse can identify references to a specific method, variable, or class names.

