Note: This provides a broad overview of the assignment. More detailed instructions will be provided once the paper is accepted.

Assignment 1: Design Antipatterns Correction (Manual Refactoring)

Description:

Locating and fixing software defects is one of the most expensive tasks involved in software development. It tends to be subjective, manual, and requires extensive knowledge of the target software design. In this case, you are assigned to fix detected defects on a given project, you will be given a set of defects types that needs to be fixed and you will need to provide a sequence of refactoring operations that will fix multiple instances of those defects. This will require analyzing the project to look for code fragments to fix (infected with smells) and then provide guidelines on how to fix them by giving a set of refactorings that you recommend. Your output will be used to optimize the quality of the software.

You are required to justify your choices in terms of refactoring decisions for each fixed defect type. You are required to fix at least 4 instances of 2 defect types, for example fix 2 God class instances, and 2 feature envy instance. Your expected number of refactoring operations to propose may be in the range of [6 ... 10].

Table 1. Refactorings catalog

Refactorings	Actors	Roles
Move method	class	source class, target class
	method	moved method
Move field	class	source class, target class
	field	moved field
Pull up field	class	sub classes, super class
	field	moved field
Pull up method	class	sub classes, super class
	method	moved method
Push down field	class	super class, sub classes
	field	moved field
Push down method	class	super class, sub classes
	method	moved method
Inline class	class	source class, target class
Extract class	class	source class, new class
	field	moved fields
	method	moved methods
Move class	package	source package, target package
	class	moved class
Extract interface	class	source classes, new interface
	field	moved fields
	method	moved methods

Assignment 2: Design Antipatterns Detection and Correction (Assisted Refactoring)

Description:
In this assignment, were interested in fixing some of them. You will choose at least 2 design flaws instances, from 2 different design flaws types (4 in total). You will use JDeodorant to come up with potential refactoring operations to fix them. Since JDeodorant sometimes gives you many recommendations on how to fix the same flaw, based on your understanding of the symptoms of design problems, choose the necessary refactorings that might solve these problems.

Task:
For this exercise, you will need to use JDeodorant. You will have JDeodorant analyze one version of a JAVA software of your choice (you can use your previous assignment project if it is feasible). You will then be asked to make some refactoring decisions and report them and their impact. Follow these steps:

1. Install Eclipse plug-in for JDeodorant
2. Run JDeodorant on a project of your choice and select 2 instances of each of the following flaws types:
    1. God Class
    2. Feature Envy
3. Now, you can look at the refactoring recommendations by JDeodorant, choose which ones to be executed, keep refactoring until you process all your chosen instances.
4. Report your findings: Chosen flaw instances, chosen refactoring operations, refactoring results. (Provide screenshots as well).
5. (Optional) Add to the report a concise comment about your experience with JDeodorant (positives, negatives and other comments).