

CSCI 2134 Lab 6: Refactoring

Fall 2023

Objective

In this lab, you will work on your own or with another student to refactor some existing code.

Preparation

1. Ensure that you have your Integrated Development Editor (IDE) installed.
2. Ensure that you are able to write and run a JUnit test case in the IDE. You will not have the time to both debug your IDE environment and complete the lab task within the lab time. The lab time should be able to concentrate on the lab task.
3. Clone the Lab 6 repo: <https://git.cs.dal.ca/courses/2023-fall/csci-2134/lab6/?????.git> where `????` is your CSID.
4. Review the documentation and provided code. These files are located in the `src` and `docs` directories of the cloned project.

Resources

- Code specifications in the `docs` directory of the Lab 6 repository.
- Code base to be refactored is in the `src` directory of the Lab 6 repository.
- Unit tests to be used for regression testing in `test` directory of Lab 6 repository.
- Lecture notes on refactoring from class.

Procedure

Set-up

1. Open the project you created in preparation for this lab
2. Open the `refactor.txt` file in `docs` and fill in your name and your partner's name (if there is one).

Part 1:

1. Ensure that the JUnit tests are passing.
2. Select a file for refactoring. We recommend one of:
 - `AngryBot.java`
 - `BullyBot.java`
 - `ChickenBot.java`
 - `SpressoBot.java`
 - `TimBot.java`
 - `DohNat.java`
 - Any other file
3. Identify a code smell in the file. **Note: You should identify at least 3 different code smells in total.** Some of them may span multiple files. For example, you should be able to find some duplicate code, magic numbers, and expressions that should be extracted into their own methods.

4. Document the code smell in the file `docs/refactor.tx`. You will need to include: (see example)
 - Name of file
 - Location (method and/or line number)
 - The identified issue
 - The proposed fix
5. Perform a refactoring to remove the code smell.
6. Rerun the tests to ensure you did not break the code. (Fix it if you do.)
7. **Commit and push to the repository. Be sure that both the code and the refactori.txt file is being committed and pushed.**
8. Repeat steps 1-7 until all opportunities are done or the lab time is completed.

Note: Only one of the partners needs to submit.

Grading

The lab will be marked out of 4 points:

Task	2 Points	1 Point	0 Points
Sufficient # of refactorings	At least 4 refactorings were completed	At least 2 refactorings were completed	Less than 2 refactorings completed
Different refactorings	At least 3 <i>different</i> refactorings were done	At least 2 different refactorings were done	Less than 2 different refactorings