

# CPSC 2151

## Lab 3

Due: Friday September 14<sup>th</sup> at 11:59 pm

In this lab you will be creating contracts for a java program.

Start by downloading the Mortgages.zip project onto your desktop (again, the one on the U drive). Unzip the folder. Open the folder in IntelliJ (run intelliJ, choose to open a project and select the folder). When you unzip you may create an additional folder called Mortgages. Make sure you select the folder that is one level above the src folder. You should now be able to work on this assignment in IntelliJ.

For this lab we'll be working on a program that a bank would use to determine whether a customer would be approved for a mortgage, and if so, what the terms of that mortgage would be. However in this lab, we will just be working on the design work, specifically the contracts, for this program. I have given you a very basic outline for two classes in this program, Customer and Mortgage. You will have to carefully consider how these classes would interact and what constraints they would have so you can write the javadoc comments, invariants, requires clauses and ensures clauses.

You do not need to write any of the code for this lab, just the Javadoc comments and all contracts.

### TIPS and additional Requirements

- You should not add any additional methods. All the methods you will need are provided.
- It's ok that we have not added any data members to these classes yet. Remember, it's better to think of the abstract concepts we know will be there, and not how exactly they are represented in data. Don't think about how the customer's credit score will be stored, just knowing that a customer's credit score will be somehow represented is enough.
- Sometimes contracts may seem obvious, especially for very simple methods, but they still need to be stated. It may be obvious what a getter function will return, but still write the contract. It may make sense how the parameters to a constructor will be mapped to the concepts of the class, but still state it in the contracts.
- Remember to start with the invariants of the class, and to use the preconditions to help you enforce the invariants by not allowing bad data in.
- Write your contracts as formally as you can
- Consider how these classes will interact and the order of function calls when considering your contracts.

### Groups

You may, but are not required to, work with a partner on this lab. Your partner must be in the same lab section as you, not just the same lecture section. If you work with a partner, only one person should submit the assignment. You should put the names of both partners in a comment at the top of the file in order for both partners to get credit. This assignment may take more than just the lab time. Make sure you are able to meet outside of class to work on the assignment before you decide to work

with someone else. Remember to actively collaborate and communicate with your partner. Trying to just divide up the work evenly will be problematic.

**Before Submitting**

Your code does not need to run on unix, since there is no code for this assignment (yet).

**Submitting your file**

You will submit your file using handin in the lab section you are enrolled in. If you are unfamiliar with handin, more information is available at <https://handin.cs.clemson.edu/help/students/>