

## WES-CS GROUP MEETING #13

### Exercise 1: Serving the Slum Lord? (Designing and Coding Classes )

A landlord owns several buildings, each of which contains several apartments. The landlord wants a program to help keep track of her rental property. In particular, she'd like to store the following information for each apartment:

- The apartment number.
- The number of bedrooms and the number of bathrooms.
- The rental price.
- Whether the apartment is currently occupied or vacant.

For each building, she'd like to store the address, plus information about each apartment in that building.

Finally, she'd like to be able to perform the following operations:

- Read current information about all buildings from a file.
- For each building, print the address of the building, and the information about each apartment.
- Change the status of an apartment (given the building address and the apartment number) from vacant to occupied or vice versa.
- Write out the updated information for all buildings to a file.

You will decide what the format of the input file will be. You will also decide the format for how the information about the buildings is be printed.

In addition to designing those formats, you will design three classes: `Apartment`, `Building`, and `ManageProperty`. The `ManageProperty` class will be run with one command-line argument: the name of the file that contains information about the landlord's buildings. It will use the information in that file to create appropriate `Apartment` and `Building` classes. The `ManageProperty` class will then allow the user to select one of the other three operations described above. When the user asks to write out the updated information, that will be done and then the program will exit.

It is also up to you to decide exactly how the `ManageProperty` class will ask the user to choose the next operation.

### **Part (a)**

First, design the `Building` and `Apartment` classes. What fields and methods should they have? Discuss your design, then write the constructors and the method that prints information about the buildings (this will let you do some testing before you write too much code).

### **Part (b)**

Now define the format of the file that stores information about buildings and apartments. Think about what will make it easiest to create the `Building` and `Apartment` classes as you read in the information. For example, you might want the file to start with the number of buildings.

### **Part (c)**

Write a simple `PropertyManager` class that has a main method that expects one command-line argument (a file name). The main method should open the file and read it, creating an array of `Buildings`. Then it should call the `Building` method that prints information. This will allow you to test the code you've written so far.

Look at Chapter 15 of the textbook, or ask your Team Leader for help with reading files if necessary.

### **Part (d)**

Finally, code and test the rest of the `Building`, `Apartment`, and `ManageProperty` classes. You may not have time to write all of the code. Plan your work so that you can test methods as you go along.