CSCI-102 Programming II - Lab 2

Create a directory named **lab2** in the **csci102/labs** directory of our shared GitHub repository. Create all of the following files in the **lab2** directory.

1. Write an Interface named **Building** that represents a physical structure.
    a. The interface should require *getters* and *setters* for the square footage (double) of the building and for the building's address (String).

2. Write a class named **CommercialBuilding** that represents a commercial building.
    a. The class should implement the **Building** interface and include fields to satisfy the methods in the interface.
    b. The class should include a field for the number of units in the building (int) along with *getter* and *setter* methods for the field.
    c. The class should set all of its fields (square footage, address, and number of units) in the constructor using arguments passed to the constructor.

3. Write an exception class named **IllFormedBuildingException** that extends the **Exception** class.
    a. The class should have a constructor that takes a String as an argument (the reason for the exception) and passes the string to the **super** class' constructor.
    b. The class overrides the *toString()* method and returns the string "IllFormedBuildingException: " concatenated with the reason for the exception.

4. Write a class named **ApartmentBuilding** that is a subclass of **CommercialBuilding**.
    a. The class has an additional field that indicates if the building has a penthouse apartment (boolean) along with a *getter* and *setter* for the field.
    b. The constructor should set all of the class fields (square footage, address, number of units, and has penthouse) using the values of the arguments passed into the constructor and by calling the superclass' constructor.
    c. The constructor should throw an **IllFormedBuildingException** if the address is null or an empty string (i.e. a string with length 0).

5. Write a program in a class named **Inventory**.
    a. Add a method named **writeApartments** that has two arguments. The first argument is a **String** holding a file name and the second argument is an **ArrayList** of **ApartmentBuilding** objects.
        i. The function writes the values all of the fields of each element in the **ArrayList** to the file named in the first argument. The file should contain one line for each object in the **ArrayList**.

    b. In main(), do the following:
        i. Create an **ArrayList** of **ApartmentBuildings**.
        ii. In a try-block, create two objects of type **ApartmentBuilding** for the following apartment buildings and put their references in the **ArrayList**.
            1. The apartment building at "35 Main Street" has 24000 square feet, 16 units, and has a penthouse apartment.
            2. The apartment building at "100 Oak Avenue" has 3000 square feet, 2 units, and does not have a penthouse apartment.
        iii. In the catch-block, catch the **IllFormedBuildingException** exception and print the reason string.
        iv. Call the *writeApartments* method, passing to the method the file name "appts.txt" and the ArrayList.