CSCI 1301- Project #2
Due on Monday, April 22nd (Monday/Wednesday class only)
Due on Tuesday, April 23rd (Tuesday/Thursday class only)

Project #2 will cover material found in the first 8 chapters of the book. I do not want you using any program features not found in the first 8 chapters of the book.

This project is broken into two separate parts/group of files. Part #1 and Part #2 are weighted the same (both are worth 50% of the project grade). I would suggest completing most or all of Part #1 before moving onto Part #2.

Part #1

If you haven't done so already, create a folder on your flash drive called CSCI1301YourLastName. Obviously you will need to put your own last name here. For example, mine would be: CSCI1301Murray

- Inside the CSCI1301 folder, I want you to create a subfolder called Project2
- Place all of your files in this folder. This will be the graded version of the project
- Name the first main file Hotel_*YourFullName*.java (replace YourFullName with your actual first and last name). Name the second file Guests.java and use the supplied text file GuestList.txt Hotel_*YourFullName*.java will contain your main method and Guests.java will contain the class that your array of objects is based on. *Note: You are not allowed to add additional files for Part #1 of the project.

This is an individual project, but you may help each other. However, I should not see any programs that are identical.

1) You will be creating a program that will assign guests to the brand new Star Wars hotel at Disney World (opening in 2020).
2) The hotel will:
    - Hold a maximum of 34 guest families
    - Your project needs to store all the guests in a 1 dimensional array of objects based on the class you create in the Guests.java file. The array needs to be big enough to handle a full hotel.
    - Each object in the array will hold information about one guest family.
        i. Full Name of the primary family member **(stored as one value)**
        ii. Street Address
        iii. City **(as a separate value from the State)**
        iv. State
        v. Number of guests in the room (including the primary family member)
        vi. The first name only of all additional guests in the room. Since there is a max occupancy of 4 per room, this may be a list of 0-3 more names.
        vii. Room# (Room number are from 101 to 134)-the program will be responsible for assigning each guest family a room. The first family will go in room 101, the second in room 102, etc. This room number will not be in the text file, your program will need to do this. I would recommend incorporating a loop counter like taking 100+1 and then 100+2, etc.

Guest input

3) No user input should be added to part 1 of the project. All 34 guest families will be read in from the file GuestList.txt. You will need to modify the text file that I supplied. I supplied you with 30 guest families. You are responsible for adding four more.
    - You are not allowed to modify the existing data in the text file. Do not add a forward slash at the end line of the last Guest (Meg Merwin).
        i. Each value is delimited from the next value with a forward slash including the end of the line. Keep this in mind when adding your 4 guest families.
        ii. Make **yourself** the first primary family member guest in the file (right above the Paul Murray line). Other than your name, you can make up the other data. I don't need your actual address or your actual family information.
        iii. After the first line…add 3 more lines with family information you make up. Paul Murray's information should now be on line #5. Make sure to leave Meg Merwin's family as the last family in the file
    - Once the data is collected, store it in an object in the array. Place the first guest family (your family) at index location 0 and work your way up. This needs to be an object based on the class you create. It cannot be a simple String array.
    - Once the hotel is full, output the full list of guests from the array to the standard output window. Do this with a for loop. One guest family per line. List only the primary person's full name, their city, their state, and their room assignment. Make sure you do this AFTER the array has been filled with the data. Do not do this as you are reading the data in from the file.

The Guests Class

The Guests Class will have the following:

1. Nine class variables for the 6-9 values collected for each object. **Note: The full name will go in one class variable, but the city and state will go in separate class variables.**
2. Create 4 constructor methods. This is accomplished via method overloading (explained in Chapter 7).
    a. One constructor method will accept 6 parameters in this order- primary family members name, street address, city, state, number of guests, room number
    b. One constructor method will accept 7 parameters in this order- primary family members name, street address, city, state, number of guests, room number, guest #2's first name
    c. One constructor method will accept 8 parameters in this order- primary family members name, street address, city, state, number of guests, room number, guest #2's first name, guest #3's first name
    d. One constructor method will accept 9 parameters in this order- primary family members name, street address, city, state, number of guests, room number, guest #2's first name, guest #3's first name, guest #4's first name

    *Note: If there isn't a guest #2 or guest #3 or guest #4 for a room, their name will remain null for the object.
3. A toString method that will print out the primary family members name, city, state, and room# for the object . Also make sure to include labels describing the data…like *Name*: Paul Murray……

Part #2

In anticipation of the new ride "Millennium Falcon: Smugglers Run" at the Star Wars: Galaxy's Edge Land opening at Disney World and Disneyland, you will create the second set of program files named Spacecraft.java and TheRide.java

- Spacecraft.java will contain a class for drawing the image/listeners/event handler methods etc. The TheRide.java file will contain the main method and the creation of the Jframe
- The second program will incorporate 2 polygon pictures you create. One polygon will be of a side view of the Millennium Falcon from the Star Wars: A New Hope movie. The second polygon will be of a side view of a Tie Fighter from the Star Wars: A New Hope movie. You may not want a true side view as the tie fighter will look like a hexagon. (Note: There is a difference between a Tie Fighter, Tie Interceptor, and Tie Bomber…but I don't care which of those 3 you choose)
- You must create the spaceships with one or more polygons. The better looking your images are, the better the design portion of your grade will be. I want you to draw this with the available polygon features found in Java. I do not want you designing it in a graphics program and importing it as a picture.
- Your program will display the Millennium Falcon so that it follows the horizontal and vertical movement of the mouse.
    i. When the left mouse button is "clicked", have landing gear appear. When it is clicked again the landing gear should disappear again. Every time the left button is pressed the landing gear should appear or disappear. Note: The Millennium Falcon does not use wheels on its landing gear.
    ii. When the right mouse button is "clicked", nothing should change
- 2 randomly moving tie fighters will move through the window. They shouldn't "follow" the mouse like the Millennium Falcon. The tie fighters can appear behind or in front of the Millennium Falcon.
    i. At least one tie fighter should be in the JFrame at all times.
    ii. Make sure these are "pseudo-randomly" moving. They shouldn't just move from one side of the JFrame to the other and they shouldn't jump or "teleport" from one spot to another completely different random spot in the window. They also shouldn't rely on mouse movement. They should be moving independently of the mouse.

Extra Credit:

Modify Part #2 so that if a tie fighter hits the Millennium Falcon, the tie fighter is destroyed and disappears. With this modification, there will be one less tie fighter randomly flying around. If you hit two, there should no longer be any tie fighters flying around. This requirement does trump the above requirement that a tie fighter should be in the JFrame at all times.

Before you turn the project in, check to see if you have the following:

1. You must have all of your project files in the one correct folder.
2. You named the first main file Hotel_*YourFullName*.java (replace YourFullName with your actual first and last name). You named the second file Guests.java and you include the updated text file GuestList.txt
3. Name the files with the polygon spacecraft: Spacecraft.java and TheRide.java
4. Proper comments including the main description block at the beginning of each file and each of the methods

The project will be graded on syntactical correctness, completeness, and quality of the graphics and graphical interaction.