Lab 10
toString, equals, Aggregation

The following exercises are to be completed during lab class. If you do not have time to finish during lab, they must be completed before the beginning of the following lab session.

Set-Up

- Create a new project in your Eclipse workspace named: **Lab10**
- In the *src* folder, create a package named: **edu.ilstu**

Part 1

Write the Java code from the `Person` class created in the pre-lab.

Write a driver (test) program named `PersonDriver` that does the following:

- Creates two `Person` objects using the custom constructor to assign values to first name, last name, and age in both objects.
- Uses the `equals` method to determine if the two objects have the same name. Messages need to use the `toString` method to print the values of each object with the appropriate messages as follows.
  - The toString should print the values of the instance variables in the form: *firstName lastName*, *age* years old. For example: Sue Smith who is 32 should print: Sue Smith, 32 years old
  - If they have the same name print:

```
Sue Smith, 32 years old and
Sue Smith, 32 years old
have the same name
```

  - If they have different names print:

```
Sue Smith, 32 years old and
George Smith, 34 years old
have different names
```

- Determines whether the two people represented by the two objects are the same age or indicates whether the first one is older or younger than the second one. Messages need to use the `toString` method to print the values of each object with the appropriate messages as follows:
  - If they are the same age

```
Sue Smith, 32 years old and
Sue Smith, 32 years old
are the same age
```

- If the first is older than the second

```
Sue Smith, 36 years old is older
George Smith, 34 years old
```

- If the first is younger than the second

```
Sue Smith, 32 years old is younger
George Smith, 34 years old
```

- Change the values assigned to the two objects and run the program again to test all of the test cases you created in pre-lab.

Part 2

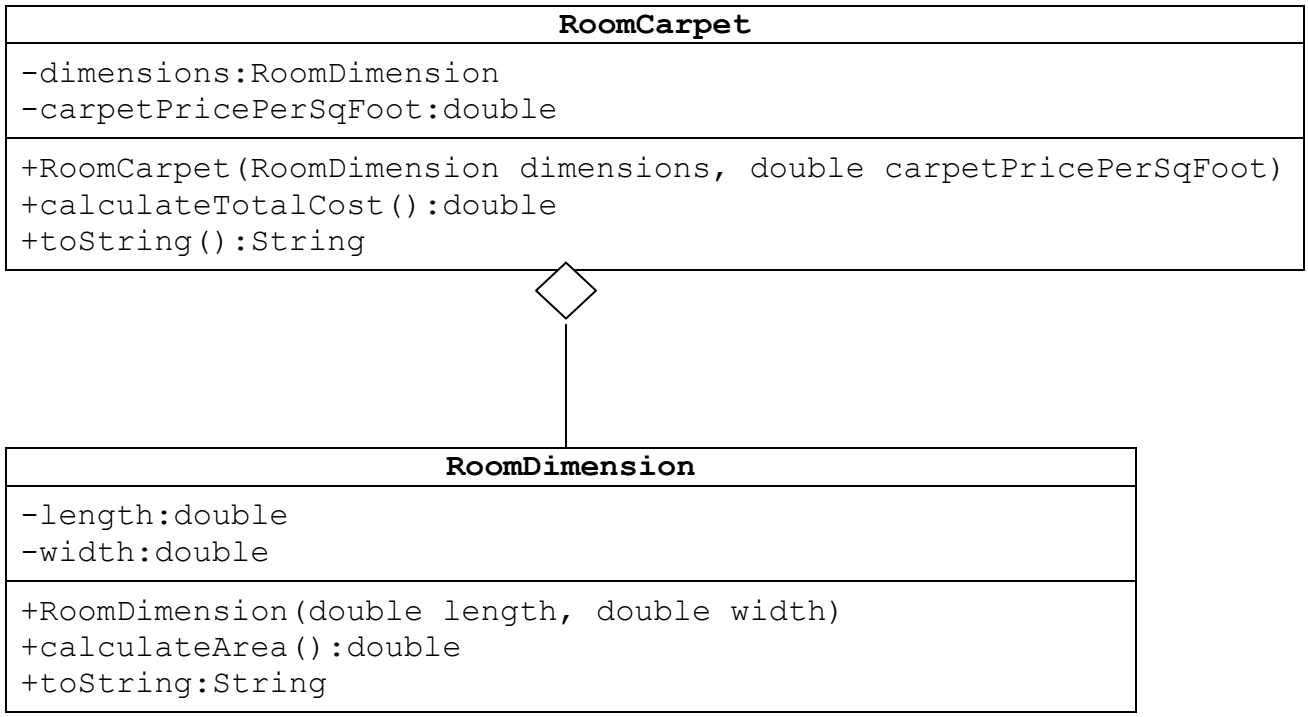
The Westfield Carpet Company has asked you to write an application that calculates the price of carpeting for rectangular rooms. To calculate the price, you multiply the area of the floor (width times length) by the price per square foot of carpet. For example, the area of a floor that is 12 feet long and 10 feet wide is 120 square feet. To cover that floor with carpet that costs $8 per square foot would cost $960. (12 x 10 x 8 = 960.)

Use the provided UML diagrams to write the code for the `RoomDimension` and `RoomCarpet` classes. The `calculateArea` method in the `RoomDimension` class returns the area of the room (the room's length multiplied by the room's width.) The `RoomCarpet` class has a `RoomDimension` object as a field. The `calculateTotalCost` method in the `RoomCarpet` class returns the total cost of the carpet. The toString for each class should include each instance variable on a separate line with a label. Do not duplicate code.

Write a driver named `CarpetDriver` using this algorithm:

CREATE `Scanner` object
PROMPT and READ length
PROMPT and READ width
PROMPT and READ price
CREATE `RoomDimension` object
CREATE `RoomCarpet` object
PRINT using the `toString` from `RoomCarpet`
CALL `calculateTotalCost`

The application should display the total cost of the carpet formatted with a dollar sign and 2 decimal places.



Sample Output

```
Enter the length of the room:  10
Enter the width of the room:  14
Enter the carpet price per square foot:  5

length = 10.0
width = 14.0
carpet cost = 5.0

Total cost =  $700.00
```

To Be Submitted

The following files should be zipped together into a file called Lab10.zip and submitted to ReggieNet by the beginning of your next lab.

- Person.java
- PersonDriver.java
- RoomCarpet.java
- RoomDimension.java
- CarpetDriver.java