CS 1400-03 Introduction to Programming and Problem Solving
Coding Practice #7
(Due: 11:59 PM, Friday, 3/19/2021)

Except Coding Practice #1, I will not grade your coding practice submissions. Instead, they will be treated as participation points. On blackboard, you will receive full points as long as you work on the exercises, which don't necessary mean they are all correct. Please check your own programs carefully and make sure they do generate the desired output.

Objectives:

- Be able to write
  - Classes, instance members, static members, constructors, getters, toString method, aggregation, copy constructors
  - Driver program to demonstrate the capabilities and features of a class
- Be able to test and debug a program

Change your working directory to `cs1400/codingPractice` for this assignment.

Task #1 Hot Dog Stands

Suppose you operate several hot dog stands distributed throughout town. Define a class named `HotDogStand` that has an instance variable for the hot dog stand's ID number and an instance variable for how many hot dogs the stand has sold that day.

- Create a constructor that allows a user of the class to initialize both values.
- Create a method named `justSold` that increments the number of hot dogs the stand has sold by one. The idea is that this method will be invoked each time the stand sells a hot dog so that you can track the total number of hot dogs sold by the stand.
- Add a `toString` method which returns the `String` value that represents the state of a `HotDogStand` object.
- Finally, add a `static` variable named `totalSold` that tracks the total number of hot dogs sold by all hot dog stands and a `static` method `getTotalSold` that returns the value of `totalSold`.

Write a driver program `HotDogStandTest.java` to test your class with three hot dog stands that each sells a variety of hot dogs. The following are **required** output from your driver program:

```
fcsang@abbott ~/cs1400/codingPractice $ java HotDogStandTest
Test our code with 3 hot dog stands created,
each sells a variety of hot dogs:
Stand 1 sold 0 hot dogs.
Stand 2 sold 0 hot dogs.
Stand 3 sold 33 hot dogs.
Total sold = 33.

Sold at stand 1, 2, 1, 1, 2:
Stand 1 sold 3 hot dogs.
Stand 2 sold 2 hot dogs.
Stand 3 sold 33 hot dogs.
Total sold = 38.
```

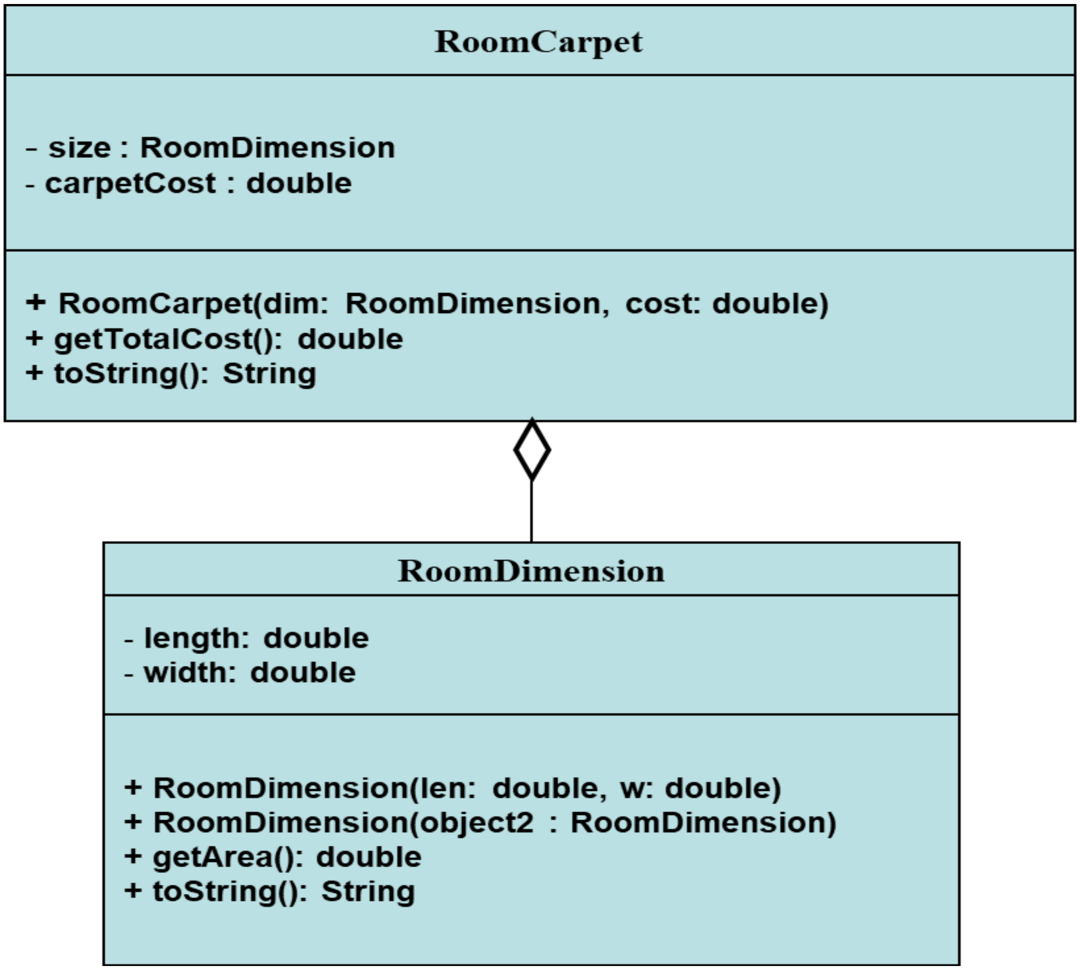
```
More sold at stand 3, 3, 2, 1, 1:
Stand 1 sold 5 hot dogs.
Stand 2 sold 3 hot dogs.
Stand 3 sold 35 hot dogs.
Total sold = 43.
```

Task #2 Carpet Calculator

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 by the price per square foot of carpet. For example, the area of 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).

First, you should create a class named `RoomDimension`. Next you should create a `RoomCarpet` class that has a `RoomDimension` object as a field. See the following UML diagram for details.

Once you have written these classes, write a driver program, called `CarpetCalculator.java`, that asks the user to enter the dimensions of a room and the price per square foot of the desired carpeting. The program should display the total cost of the carpet.



Below is a sample execution of the program.

```
fcsang@abbott ~/cs1400/codingPractice $ java CarpetCalculator
This program will display price to carpet a room.
Enter the length of the room: 10
Now enter the width: 12
Enter carpet price per square foot: 8

Room dimension:
Length = 10.0
Width = 12.0
Carpet cost per square foot: $8.0
Total cost: $960.0
```

Submission:
Generate a `script` file `practice7.txt` with appropriate time stamps and the following steps visible:

1) a `pwd` to show the current working directory
2) a `ls -l` to show in long format the files in your `cs1400/codingPractice` directory
3) display both `HotDogStand.java` and `HotDogStandTest.java`
4) compile `HotDogStandTest.java`
5) run `HotDogStandTest`
6) display `RoomDimension.java`, `RoomCarpet.java` and `CarpetCalculator.java`
7) compile `CarpetCalculator.java`
8) run `CarpetCalculator`

Submit the script file `practice7.txt` on Bb, under the Coding Practice Folder, Practice #7 link.