

CSE 142 - Su 02

Homework 5

Assigned: Wednesday, July 31

Due: Wednesday, August 7, BEFORE MIDNIGHT

**** General Comments about the Homework ****

All homework is turned in electronically. Go to the class web site and use the link on the homework page to do the turnin. Don't be late! Late homeworks will not be accepted.

This file describes the Homework 5 Graded Problems. There are other files that describe the practice problems and the programming project.

**** Homework 5 Graded Problems ****

Each of the following 5 problems is worth 2 points each.

1. Consider the code in Metro.java. The only method in that class is the static void main method. This is where execution of the whole program starts. For this question, I want you to trace the flow of control after the program starts while it runs through the various constructors and methods.

The program starts execution with the first line of the main method. It creates a TransitSystem object and several buses. It prints some information about the buses.

On line 22, main calls the Landmark constructor. Go read the Landmark constructor. What does it do? It calls the Location constructor. What does the Location constructor do? It calls the update method of the Location class. The update method might call the IllegalArgumentException constructor, but we'll assume that it doesn't since that only happens when there is an error. Update finishes and returns to the Location constructor. The Location constructor finishes and returns to the Landmark constructor. The Landmark constructor finishes and return to main in class Metro. Now we're ready for the next line of main!

The preceding sequence of control can be summarized in a table like this.

Class name	Constructor or method name
Metro	main
Landmark	constructor
Location	constructor
Location	update

In the same fashion as above (text followed by a summary table) describe the sequence of method calls that takes place when the program executes line 25 in the main method:

```
System.out.println(kc.findClosestBus(hub));
```

You should specify the calls to the ArrayList methods like get() and size(), but you don't have to follow them to see what they do. When things happen in a loop, you can just say "this sequence of statements repeats for each entry in the list" rather than repeating them over and over again.

2. We have used ArrayLists in package java.util quite a bit. There are numerous other predefined data structures in the same package that are also very useful.

a. Consider the documentation for `java.util.TreeSet`. The top of the inheritance chain is class `Object`, as it is for all Java classes. Write down all the classes that lead from `Object` to `TreeSet`.

b. Consider the documentation for `java.util.ArrayList`. The top of the inheritance chain is class `Object` again. Write down all the parent classes that are shared by `ArrayList` and `TreeSet`.

3. Consider the documentation for the `Triangle` class in the `uwscse.graphics` package.

a. If you have an object of class `Triangle`, how many different method names can be used with that object?

b. Of those methods, how many are actually defined in the `Triangle.java` file? (You can tell from the documentation, you don't need the sources.)

4. Write a complete java source file that defines an interface called `GeometricFigure`, which will be used to describe various geometric figures (circles, squares, and so on). This interface defines two public methods:

`getArea`, which takes no arguments and returns a `double` (which is the area of the figure).

`isBig`, which takes no arguments and returns a `boolean` (`true` if the area of the figure is > 1 , `false` otherwise).

Be sure to include appropriate javadoc comments in your code.

5. Write a complete java source file that defines a class called `MyCircle` that implements `GeometricFigure` (described in the previous question).

Include the following in your implementation:

Write a `MyCircle` constructor that takes a value of type `double`, and sets an instance variable named `radius` to that value.

Write the methods required by the `GeometricFigure` interface. You can calculate the area of the circle as $\text{Math.PI} * \text{radius} * \text{radius}$.

Write two additional public methods:

`setRadius`, which takes one argument of type `double`, sets the `radius` instance variable, and returns nothing.

`getRadius`, which takes no arguments and returns the `radius` of the circle.

Be sure to include appropriate javadoc comments in your code.