Java Strings, Classes and Interfaces- Hands-On Exercises

Assignment 01  
Write a program to give examples of operators.

1. Increment and decrement operators.
2. Arithmetic operator.
3. Relational Operator
4. Conditional Operator.

Assignment 02  
Setting up Classes with Interfaces

Assume you want to capture shapes, which can be either circles (with a radius and a color) or rectangles (with a length, width, and color). You also want to be able to create signs (to post in the campus center, for example), each of which has a shape (for the background of the sign) and the text (a String) to put on the sign.

Create classes and interfaces for circles, rectangles, shapes, and signs.

In order to make signs, we need to make shapes that are large enough to fit the text for the sign. Write a method on shapes fitsText that takes the text as an argument and determines whether the length of the text is shorter than the length/radius of the shape. You can get the length of a String by calling the method length on the string.

Practice making errors with your shapes: omit the interface, make typos in the names of the fitsText method, and so on. This will help you get familiar with the errors messages that Java produces in various situations.

### Understanding Interfaces

Here’s a basic setup of classes and an interface for vehicles. Use this to answer the following question about interfaces.

|  |
| --- |
| interface IVehicle { |
| // indicate how much a basic tune-up costs |
| public double tuneUpCost(); |
|  |
| // determine whether vehicle can hold given num of passengers |
| public boolean canCarry(int numPassengers); |
| } |
|  |
| class Car implements IVehicle { |
| int mileage; |
| int year; |
| int numDoors; |
|  |
| // constructor goes here |
|  |
| // indicate whether car was built before given year |
| boolean builtBefore(int otherYear) { |
| return this.year < otherYear; |
| } |
| } |
|  |
| class Bicycle implements IVehicle { |
| int mileage; |
| int numGears; |
|  |
| // constructor goes here |
| } |

1. What methods do you need to add to each of Car and Bicycle to get this code fragment to compile (setting aside the missing constructors)?
2. Does having a class implement an interface change how you write its constructor?
3. Should builtBefore be added to the IVehicle interface? Why or why not?
4. In the Examples class, you want to define a Bicycle as follows:

|  |
| --- |
| \_\_\_\_\_\_\_\_\_\_\_ newKidsBike = new Bicycle(0, 1); |

Which types can you use in the blank line (without causing a compile error just with this single line)?

1. Assume you defined a Car object as follows:

|  |
| --- |
| IVehicle oldCar = new Car(200000, 1995, 2); |

Which methods can you call on oldCar, given the type that you gave it?

1. Assume you defined a Car object as follows:

|  |
| --- |
| Car oldCar = new Car(200000, 1995, 2); |

Which methods can you call on oldCar, given the type that you gave it?

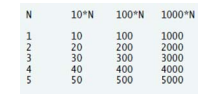
Assignment 03

The process of finding the largest value (i.e., the maximum of a group of values) is used frequently in computer applications. For example, a program that determines the winner of a sales contest would input the number of units sold by each sales person. The sales person who sells the most units wins the contest. Write a Java application that inputs a series of 10 integers and determines and prints the largest integer. Your program should use at least the following three variables:

1. **counter**: A counter to count to 10 (i.e., to keep track of how many numbers have been input and to determine when all 10 numbers have been processed).
2. **number**: The integer most recently input by the user.
3. **largest**: The largest number found so far.

Assignment 04

Write a Java application that uses looping to print the following table of values:



Assignment 05

Create a class called Employee that includes three pieces of information as instance variables—a first name (type String), a last name (type String) and a monthly salary (double). Your class should have a constructor that initializes the three instance variables. Provide a set and a get method for each instance variable. If the monthly salary is not positive, set it to 0.0.

1. Write a test application named EmployeeTest that demonstrates class Employee’s capabilities.
2. Create two Employee objects and display each object’s yearly salary.
3. Then give each Employee a 10% raise and display each Employee’s yearly salary again.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*