Alvin Tran

Assignment 2

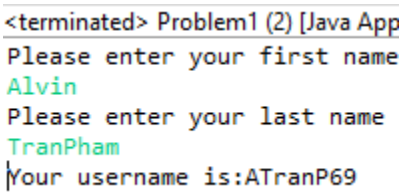
6/12/2021

Problem 1

Problem 1 requires a user prompt asking for 2 strings, First and Last name, and then returns a string that is a combination of the 1st letter of First name, first 5 letters of Last name, and then a random number between 10 and 99.

Firstly, the problem requires the **java.util.Scanner** class to read in the two strings using the **.next()** method. Second, we need to use **.substring()** method to get the appropriate letters from the First and Last name, which would be [0-1] and [0-5] respectively for their substrings, Third, we must utilize the **math.random()** function to get a random number. **Math.random()** outputs a number between 0 and 1 inclusive, so we multiply it by 89 so we get a number from 0-89, then add 10 to get our 10-99 range.

Finally, we output the concatenation of the data we retrieved above, which we can simply use by using the **(+) plus** operator.

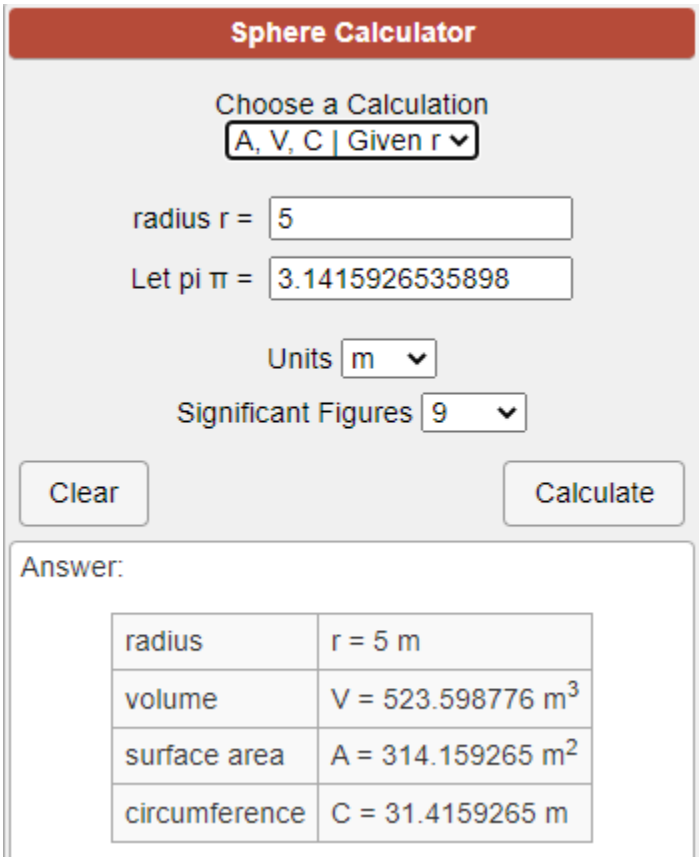


Example Input and Output

Problem 2

Problem 2 wants us to calculate Volume and Surface Area of a sphere, and the output it. The radius is prewritten into the program since we do not need to prompt the user. Furthermore, we must ensure our output is formatted to 4 decimal places.

We simply utilize the **Math.PI** and **Math.pow()** lines to calcuilate volume and surface are. More importantly though, we import and use the **DecimalFormat** class with the input of **"###.####".** This limits the decimal to 4 past the decimal, since we put 4 #'s in out string. We then use the **.format()** method of our formatter to format out Volume and Surface Area doubles.



Expected Output

```
<terminated> Problem2 (1) [Java Application]
For a sphere of radius: 5
Its volume is: 523.5988
Its surface area is: 314.1593
```

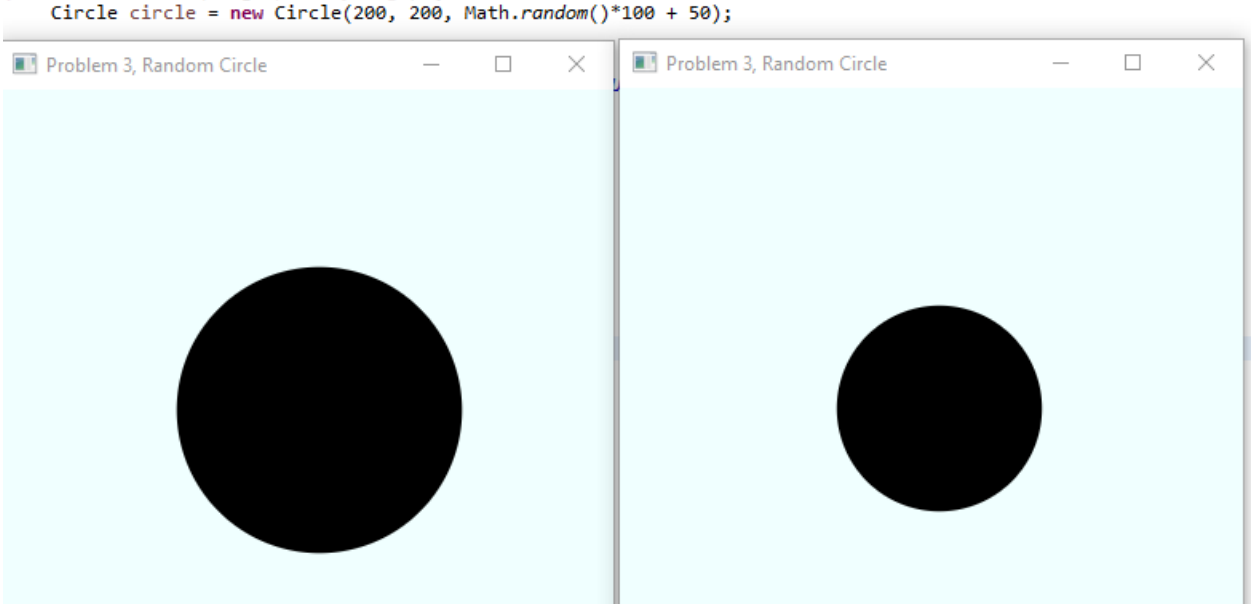
Output Given

Problem 3

Problem 3 wants us to create a **javafx** program to make a circle appear at (200, 200) of the window with a random radius from 50 to 150 pixels.

We use the **Circle** class from the **javafx** library to initialize our Circle, its position, and radius. Its radius is determined by **Math.random(),** which we multiply by 100 (our range) and add 50 (our minimum) to get 50-150 as our possible radii. We appropriately add the circle into our **Group, Scene,** and **Stage** to display.

```
Circle circle = new Circle(200, 200, Math.random()*100 + 50);
```



2 Different Runs of Same Code

Problem 4

Problem 3 requires a c named **Flight** which will contain instance data of Airline name, Flight number, Origin City, and Destination City. Each variable will have a get/set method. A constructor must be made for Flight that will set all the data when we create a Flight. In addition, we need a **toString()** method that will output all our data in 1 line for the user. Not only that, be we must prove the functionality of our methods.

I created a separate .java file named **Flight.java** to be utilize with our main file. We create private variables that represent the data we need named above. We write **getX()** methods by using the **return** keyword to pass X variable, and **setY()** methods by giving it an argument of matching datatype then assigning the input argument to the instance variable. The **constructor** likewise is created as we did with the **setY()** methods, but conjoined together. Finally, we make a method **toString()** which uses **return** to return a string we create using the private variables and hard coded strings for user readability.

```
Flight flight1 = new Flight("Southwest", 1, "Louisvlle", "Chicago");
Flight flight2 = new Flight("United", 2, "New York", "Florida");
Flight flight3 = new Flight("Delta", 0, "Unkown", "Unknown");
```

Initial Objects Created

```
System.out.println(flight1.toString());

//Flight 2 changed Airlines
flight2.setName("Southwest");
System.out.println("Flight 2's airline has been changed to " + flight2.getName());
System.out.println(flight2.toString());

//Flight 3 gets initialized
flight3.setNumber(3);
flight3.setOrigin(flight1.getDestination());
flight3.setDestination(flight1.getOrigin());
System.out.println("Flight 1's return flight will be flight 3");
System.out.println(flight3.toString());
```

Functionality Tested

As shown above, we expect Flight 1 to output exactly what we initialized it before. We also expect Flight 2's airline to be changed to Southwest using **setName(),** and the **getName()** and **toString()** to verify that action. For flight 3, use the **getDestination()** and **getOrigin()** our flight 1 to make a return flight by swapping origin and destination appropriately with the set functions. We also give **flight3** the flight number 3.

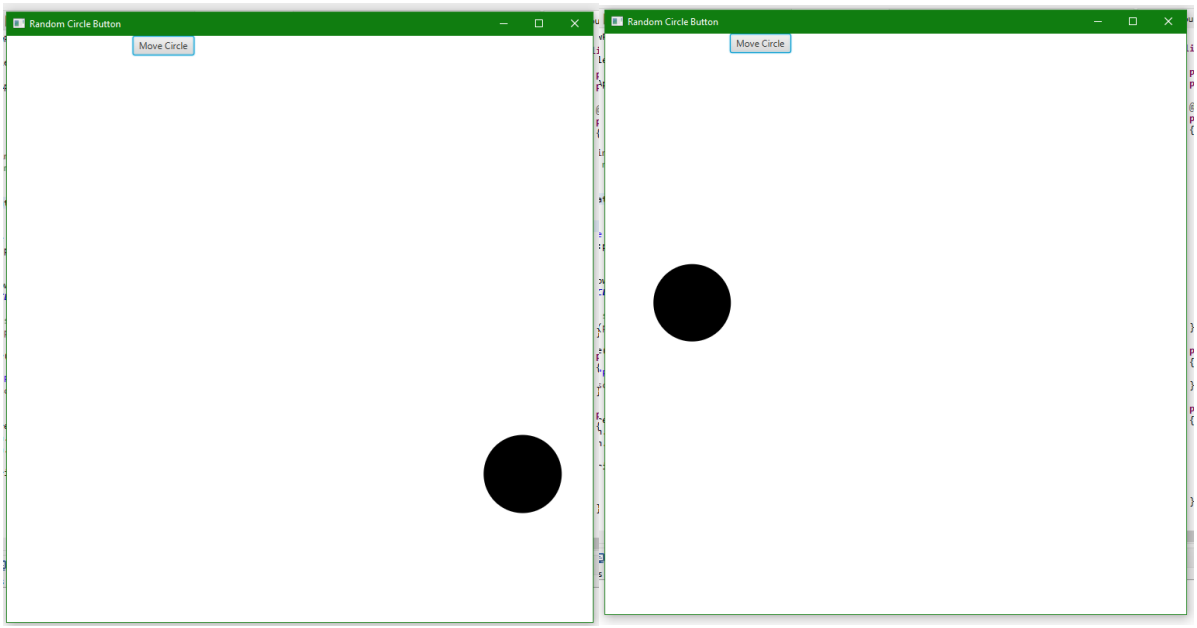
```
Airline Name: Southwest, Flight Number: 1, Origin: Louisvlle, Destination: Chicago


Airline Name: United, Flight Number: 2, Origin: New York, Destination: Florida
Flight 2's airline has been changed to Southwest
Airline Name: Southwest, Flight Number: 2, Origin: New York, Destination: Florida


Airline Name: Delta, Flight Number: 0, Origin: Unkown, Destination: Unknown
Flight 1's return flight will be flight 3
Airline Name: Delta, Flight Number: 3, Origin: Chicago, Destination: Louisvlle
```

Matching Output

Problem 5

Problem 5 requires us to create JavaFX application with a button and circle, in which the button randomizes the position of the circle.

We solve this problem by first creating a **Button** and a **Circle** object. We then create variables xBound and yBound to dictate Scene size and the range in which our circle may appear. We then create a **method** that processes the **event** of button press. Inside it, we simply use **setCenterX()** and **setCenterY()** to and **Math.random()** with out bounds we created to reinstate the position of our circle. We appropriately link that method the button with **setOnAction()** and the **this::** operator. We then appropriately put everything into a group, scene, and stage as we would any other JavaFX program.



Same Run of Code, Before and After Press