Unit 5 Programming Problems Worksheet

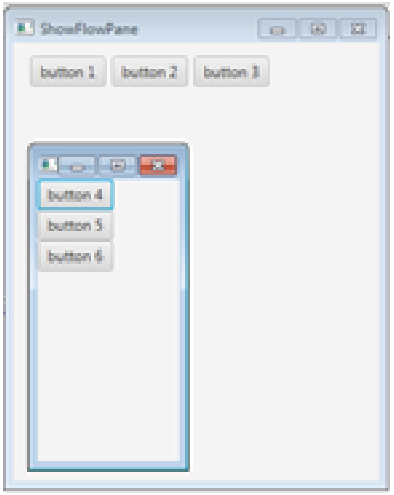
Programming Problem 1 – FlowPane

Write a program that meets the following requirements.

- Create two stages in a program
- Create a pane using FlowPane in each stage
- Add three buttons to each pane

Directions

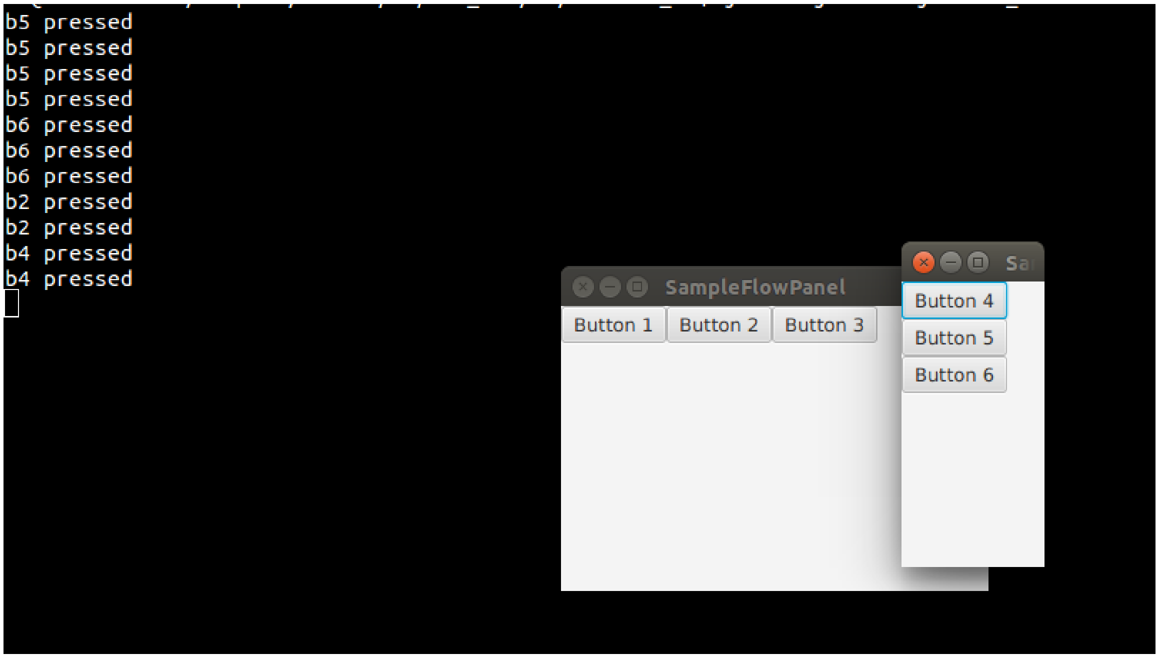
- Create a class named FlowPaneDemo extends Application
- Create user interface using FlowPane
- Add the instances of 3 Buttons to pane1 created by FlowPane and other 3 instances of Buttons using FlowPane to pane2
- Create scene1 for pane1 with a specific size and scene2 for pane2 with a different size
- Set different titles to two stages and display two stages
- The output should look like the screen below
- Provide appropriate Java comments



Grading Rubric

Task	Points
Create user interface using Button, FlowPane and Stage	2
Add instances of 3 Buttons using FlowPane named pane1, and 3 other instances of Buttons using FlowPane named pane2	2
Add pane1 and pane 2 to scene1 and scene2	2
Set different titles to stages and different sizes to scenes	2
The output for FlowPane is correct and looks like the sample screen	2
Total	10

Screenshots



```
b5 pressed
b5 pressed
b5 pressed
b5 pressed
b6 pressed
b6 pressed
b6 pressed
b2 pressed
b2 pressed
b4 pressed
b4 pressed
```

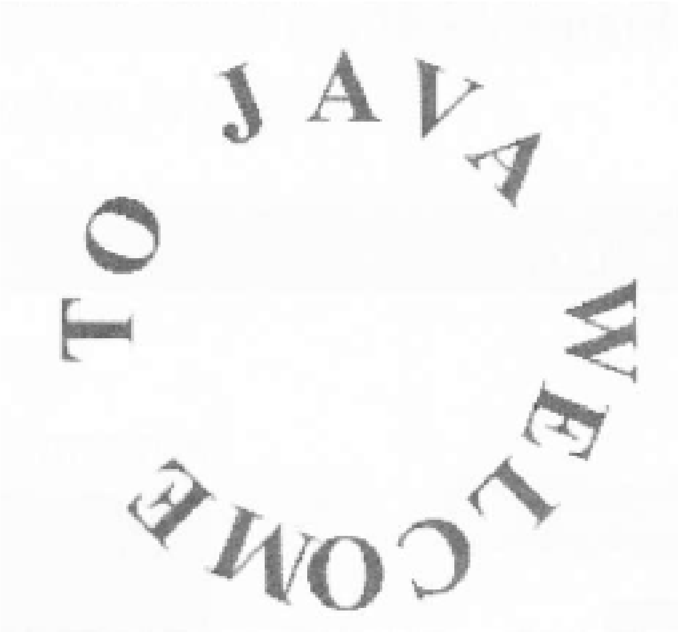
Programming Problem 2 – Characters Around a Circle

Write a program that displays a string “Welcome to Java” around the circle, as shown below. You need to display each character in the right location with appropriate rotation using a loop.

Directions

- Create a class named Characters extends Application.
- Create a new pane.

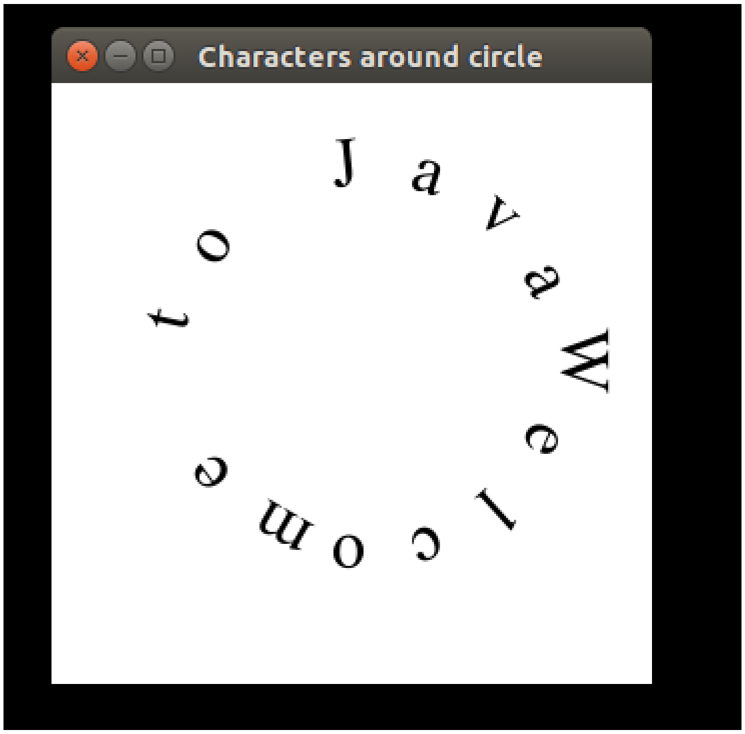
- Create an object of Font class and define font features as bold, Times New Roman, regular and size of 35.
- Create a string of “Welcome to Java”
- Create a for loop to scan through each character of “Welcome to Java” string.
- Inside the for loop, create an instance of Text on the given coordinates containing the given characters by use constructor of Text(double x, double y, java.lang.String text)
- Use setFont( ) and setRotate( ) methods of Text class to define characters and their rotated positions
- Add text instance to the pane by using getChildren().add() method
- Create a scene with a specific size
- Set title to “Characters around circle” and display the stage
- The output should look like the screen below.
- Provide appropriate Java comments



Grading Rubric

Task	Points
Create user interface using Scene, Pane and Stage	2
Use Font class correctly to set up font style and size	3
Use for loop correctly to scan through each character of the string	2
Use methods of Text class to set font and rotate characters	3
The output for Characters Around Circle is correct and looks like the sample screen	2
Total	**12**

Screenshots



Programming Problem 3 – Shuffling Cards

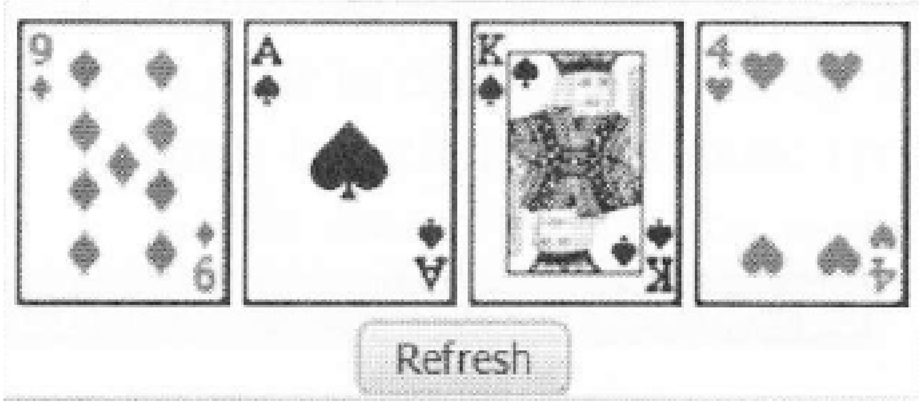
Write a program that displays four cards randomly selected from a deck of 52 if the Refresh button is clicked, shown below. The card image files are named 1.png, 2.png … 52.png and stored in the image/card directory. All four cards are distinct and selected randomly. The card files are provided in Blackboard.

Hint: You can select random cards by storing the numbers 1-52 to an array list, perform a random shuffle learned in Section 11.12, and use the first four numbers in the array list as the file names for the image. The other way is to use the static shuffle method in the java.util.Collections class, for example: java.util.Collections.shuffle(list) where list is an array list.

Directions

- Create a class named CardRefreshButton extends Application.
- Create a new array list, then use a for loop and add() method to add 52 cards to the list.
- Revoke shuffle() method for the list to shuffle the cards.
- Create an instance of HBox and add four card images by using getChildren().add() method. Make sure specify image file path and use array list's get() method to get four card image files.
- Create a Button and labeled with "Refresh"
- Create and register an event handler.
- To handle event: once the button is clicked, call shuffle() method to shuffle the cards, then clear previous images by using getChildren.clear() method, and use add() to add four card images.
- Create a new BorderPane to set up layout for the images and button.

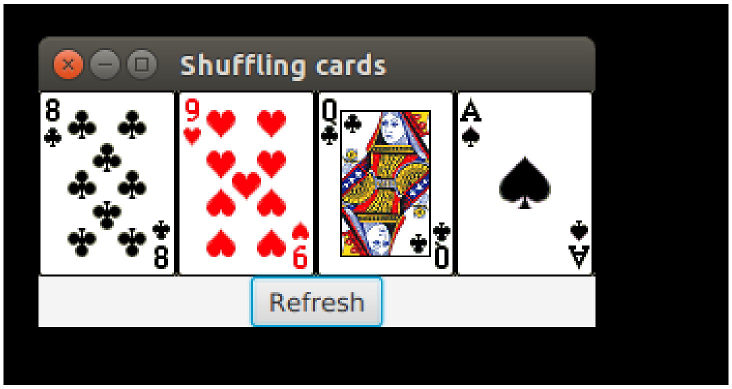
- Create a scene with a specific size.
- Set title to “Shuffling cards” and display the stage.
- The output should look like the screen below.
- Provide appropriate Java comments.

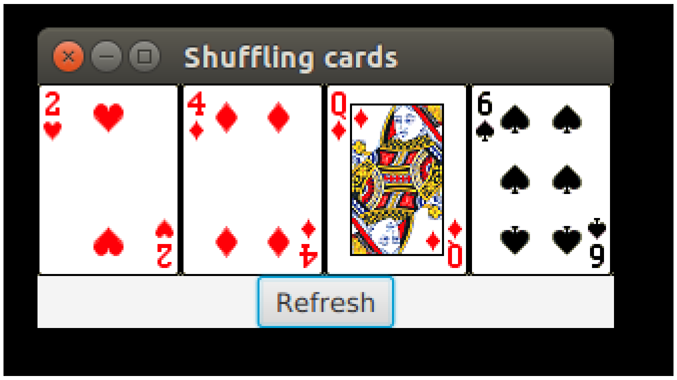


Grading Rubric

Task	Points
Create user interface using Scene, Pane and Stage	2
Use array list and a for loop to hold 52 cards	2
Shuffle cards by using random selection or static shuffle method	2
Create a HBox and use appropriate methods to add images and setup layout	3
Create Refresh button and handler	2
Use methods to shuffle, clear and add images correctly to handle button click event	3
Use BorderPane correctly to setup layout	2
The output for Shuffling Cards is correct and looks like the sample screen	2
Total	**18**

Screenshots


Shuffling cards
Refresh


Shuffling cards
Refresh