Lab-10
Exploring JavaFx

Objectives:
Show different JavaFx Layouts and Charts.

Theory:
JavaFx Layouts

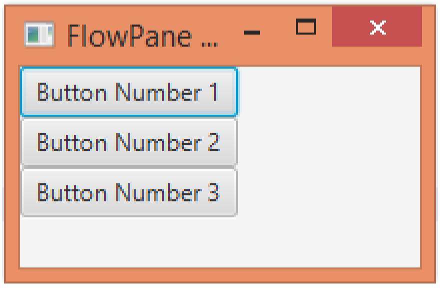
Layouts are the top level container classes that define the UI styles for scene graph objects. In JavaFX, Layout defines the way in which the components are to be seen on the stage. It basically organizes the scene-graph nodes. We have several built-in layout panes in JavaFX that are HBox, VBox, StackPane, FlowBox, AnchorPane, etc. Each Built-in layout is represented by a separate class which needs to be instantiated in order to implement that particular layout pane.
All these classes belong to **javafx.scene.layout** package. **javafx.scene.layout.Pane** class is the base class for all the built-in layout classes in JavaFX.

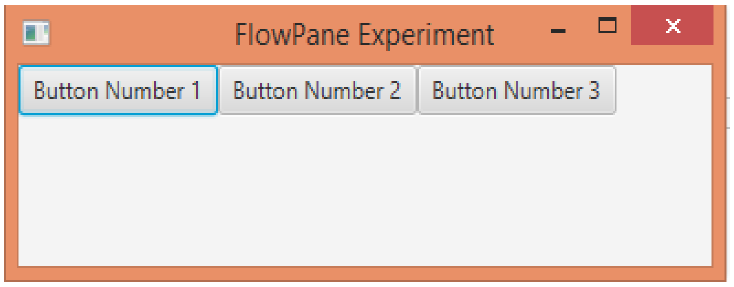
Class	Description
BorderPane	Organizes nodes in top, left, right, centre and the bottom of the screen.
FlowPane	Organizes the nodes in the horizontal rows according to the available horizontal spaces. Wraps the nodes to the next line if the horizontal space is less than the total width of the nodes
GridPane	Organizes the nodes in the form of rows and columns.
HBox	Organizes the nodes in a single row.
Pane	It is the base class for all the layout classes.
StackPane	Organizes nodes in the form of a stack i.e. one onto another
VBox	Organizes nodes in a vertical column.

FlowPane

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.layout.FlowPane;
import javafx.stage.Stage;
public class FlowPaneExperiments extends Application  {
   @Override
   public void start(Stage primaryStage) throws Exception {
      primaryStage.setTitle("FlowPane Experimwnt");
      Button button1 = new Button("Button Number 1");
      Button button2 = new Button("Button Number 2");
      Button button3 = new Button("Button Number 3");
      FlowPane flowpane = new FlowPane();
      flowpane.getChildren().add(button1);
      flowpane.getChildren().add(button2);
```

```
        flowpane.getChildren().add(button3);
        Scene scene = new Scene(flowpane, 200, 100);
        primaryStage.setScene(scene);
        primaryStage.show();
    }
    public static void main(String[] args) {
        Application.launch(args);
    }
}
```

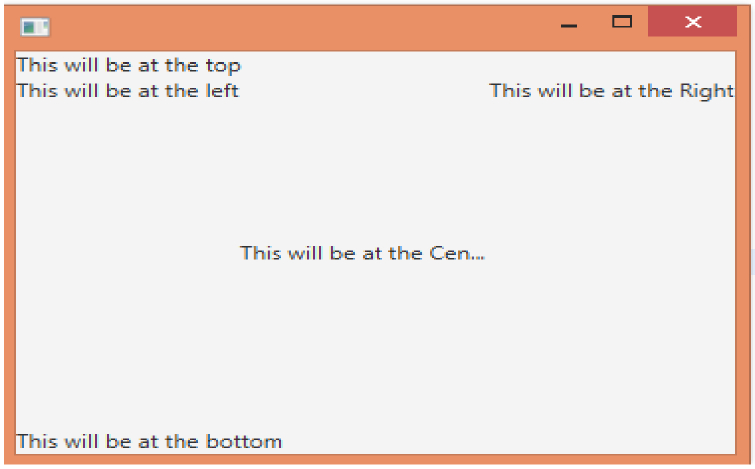




BorderPane

```
package application;
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Label;
import javafx.scene.layout.*;
import javafx.stage.Stage;
public class Label_Test extends Application {
    @Override
    public void start(Stage primaryStage) throws Exception {
        BorderPane BPane = new BorderPane();
        BPane.setTop(new Label("This will be at the top"));
        BPane.setLeft(new Label("This will be at the left"));
        BPane.setRight(new Label("This will be at the Right"));
        BPane.setCenter(new Label("This will be at the Centre"));
        BPane.setBottom(new Label("This will be at the bottom"));
        Scene scene = new Scene(BPane,600,400);
        primaryStage.setScene(scene);
        primaryStage.show();
    }
    public static void main(String[] args) {
        launch(args);
    }

}
```



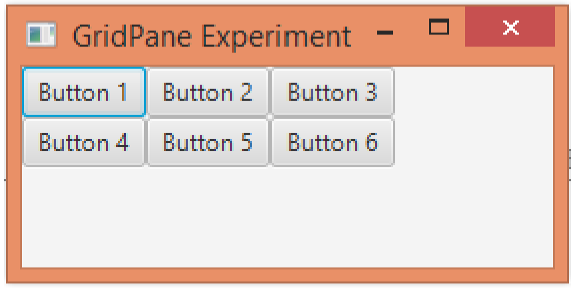
```
This will be at the top
This will be at the left                     This will be at the Right

                  This will be at the Cen...

This will be at the bottom
```

GridPane

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.layout.GridPane;
import javafx.stage.Stage;
public class GridPaneExperiments extends Application  {
   @Override
   public void start(Stage primaryStage) throws Exception {
      primaryStage.setTitle("GridPane Experiment");
      Button button1 = new Button("Button 1");
      Button button2 = new Button("Button 2");
      Button button3 = new Button("Button 3");
      Button button4 = new Button("Button 4");
      Button button5 = new Button("Button 5");
      Button button6 = new Button("Button 6");
      GridPane gridPane = new GridPane();
      gridPane.add(button1, 0, 0);
      gridPane.add(button2, 1, 0);
      gridPane.add(button3, 2, 0);
      gridPane.add(button4, 0, 1);
      gridPane.add(button5, 1, 1);
      gridPane.add(button6, 2, 1);
      Scene scene = new Scene(gridPane, 240, 100);
      primaryStage.setScene(scene);
      primaryStage.show();
   }
   public static void main(String[] args) {
      Application.launch(args);
   }
}
```



JavaFx Charts

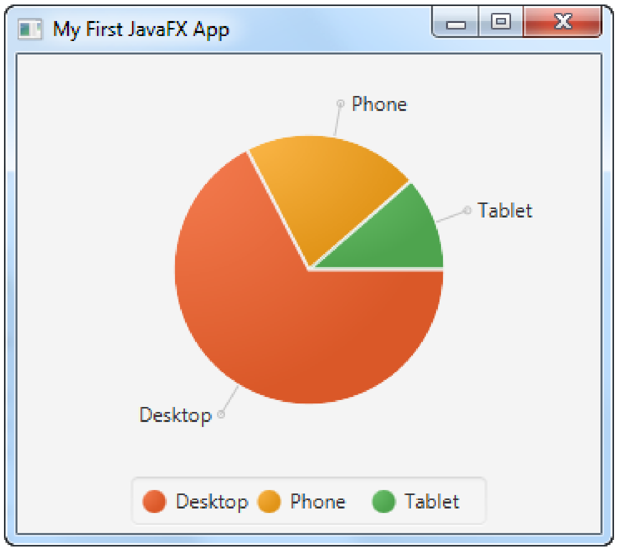
In general, a chart is a graphical representation of data. There are various kinds of charts to represent data such as **Bar Chart, Pie Chart, Line Chart, Scatter Chart,** etc.

JavaFX Provides support for various **Pie Charts** and **XY Charts**. The charts that are represented on an XY–plane include **AreaChart, BarChart, BubbleChart, LineChart, ScatterChart, StackedAreaChart, StackedBarChart,** etc.

Each chart is represented by a class and all these charts belongs to the package **javafx.scene.chart**. The class named **Chart** is the base class of all the charts in JavaFX and the **XYChart** is base class of all those charts that are drawn on the XY–plane.

// Code for Pie Chart

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.chart.PieChart;
import javafx.scene.layout.VBox;
import javafx.stage.Stage;
public class PieChartExperiments extends Application {
   @Override
   public void start(Stage primaryStage) throws Exception {
      primaryStage.setTitle("My First JavaFX App");
      PieChart pieChart = new PieChart();
      PieChart.Data slice1 = new PieChart.Data("Desktop", 213);
      PieChart.Data slice2 = new PieChart.Data("Phone"  , 67);
      PieChart.Data slice3 = new PieChart.Data("Tablet" , 36);
      pieChart.getData().add(slice1);
      pieChart.getData().add(slice2);
      pieChart.getData().add(slice3);
      VBox vbox = new VBox(pieChart);
      Scene scene = new Scene(vbox, 400, 200);
      primaryStage.setScene(scene);
      primaryStage.setHeight(300);
      primaryStage.setWidth(500);
      primaryStage.show();
   }
   public static void main(String[] args) {
        Application.launch(args);
    }
}
```



<u>Lab Task:</u>
Rewrite all programs

- place your name in the labels in Border pane
- place your name in buttons of Flowpane
- place your name in buttons in Gridpane
- place your name and your friends name in each slice of PieChart

<u>Conclusion:</u>