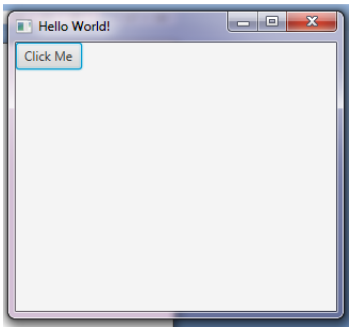
Layout

1) **Show Button Using Hbox**

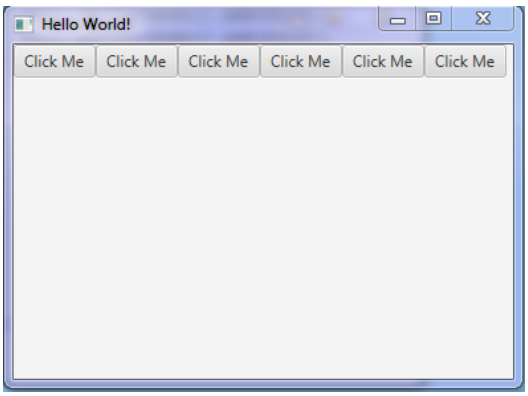
```
package layoutexamplehbox;

import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.stage.Stage;
import javafx.scene.layout.HBox;

public class LayoutExampleHbox extends Application {

   @Override
   public void start(Stage primaryStage) {
      Button btn = new Button("Click Me");
      HBox root=new HBox();
      root.getChildren().add(btn);
      Scene scene = new Scene(root, 300, 250);

      primaryStage.setTitle("Hello World!");
      primaryStage.setScene(scene);
      primaryStage.show();
   }

   public static void main(String[] args) {
      launch(args);
   }

}
```



2) **Program to display working of HBOX**

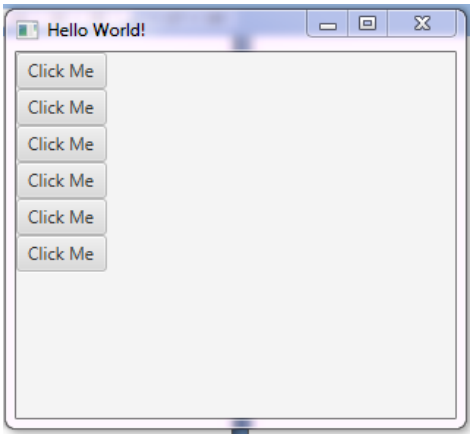
```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.stage.Stage;
```

```
import javafx.scene.layout.HBox;

public class LayoutExampleHbox extends Application {

   @Override
   public void start(Stage primaryStage) {
      Button btn1 = new Button("Click Me");
      Button btn2 = new Button("Click Me");
      Button btn3 = new Button("Click Me");
      Button btn4 = new Button("Click Me");
      Button btn5 = new Button("Click Me");
      Button btn6 = new Button("Click Me");

      HBox root=new HBox();
      root.getChildren().add(btn1);
      root.getChildren().add(btn2);
      root.getChildren().add(btn3);
      root.getChildren().add(btn4);
      root.getChildren().add(btn5);
      root.getChildren().add(btn6);

      Scene scene = new Scene(root, 300, 250);

      primaryStage.setTitle("Hello World!");
      primaryStage.setScene(scene);
      primaryStage.show();
   }

   public static void main(String[] args) {
      launch(args);
   }

}
```



3) Show Button Using Vbox

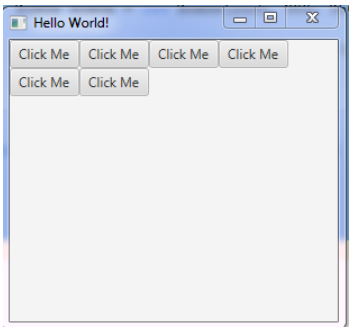
```
package layoutexamplehbox;

import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.stage.Stage;
import javafx.scene.layout.VBox;

public class LayoutExampleHbox extends Application {

    @Override
    public void start(Stage primaryStage) {
        Button btn1 = new Button("Click Me");
        Button btn2 = new Button("Click Me");
        Button btn3 = new Button("Click Me");
        Button btn4 = new Button("Click Me");
        Button btn5 = new Button("Click Me");
        Button btn6 = new Button("Click Me");

        VBox root=new VBox();
        root.getChildren().add(btn1);
        root.getChildren().add(btn2);
        root.getChildren().add(btn3);
        root.getChildren().add(btn4);
        root.getChildren().add(btn5);
        root.getChildren().add(btn6);

        Scene scene = new Scene(root, 300, 250);

        primaryStage.setTitle("Hello World!");
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    public static void main(String[] args) {
        launch(args);
    }

}
```



4) Show Button Using FlowPane

```
package layoutexamplehbox;

import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.layout.FlowPane;
import javafx.stage.Stage;

public class LayoutExampleHbox extends Application {

   @Override
   public void start(Stage primaryStage) {
      Button btn1 = new Button("Click Me");
      Button btn2 = new Button("Click Me");
      Button btn3 = new Button("Click Me");
      Button btn4 = new Button("Click Me");
      Button btn5 = new Button("Click Me");
      Button btn6 = new Button("Click Me");

      FlowPane root=new FlowPane();
      root.getChildren().add(btn1);
      root.getChildren().add(btn2);
      root.getChildren().add(btn3);
      root.getChildren().add(btn4);
      root.getChildren().add(btn5);
      root.getChildren().add(btn6);

      Scene scene = new Scene(root, 300, 250);

      primaryStage.setTitle("Hello World!");
      primaryStage.setScene(scene);
```

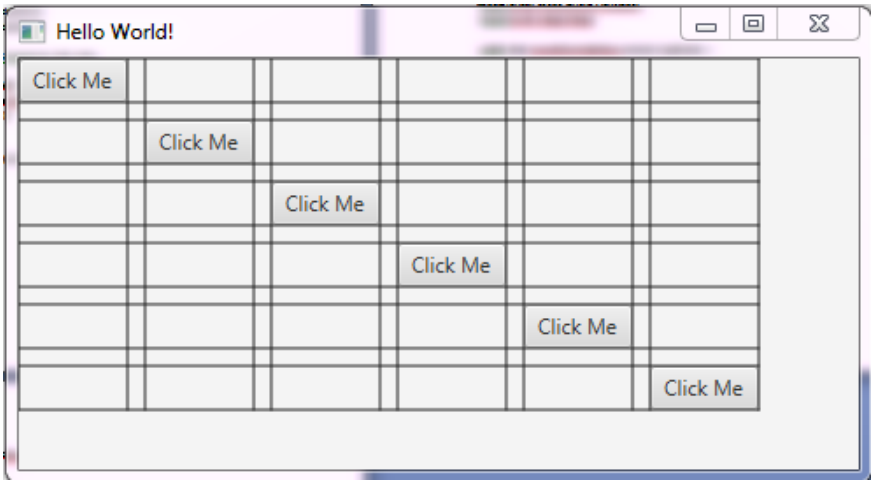
```
        primaryStage.show();
    }

    public static void main(String[] args) {
        launch(args);
    }

}
```



5) Show Button Using GridePane

```
package layoutexamplehbox;

import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.layout.GridPane;
import javafx.stage.Stage;

public class LayoutExampleHbox extends Application {

    @Override
    public void start(Stage primaryStage) {
        Button btn1 = new Button("Click Me");
        Button btn2 = new Button("Click Me");
        Button btn3 = new Button("Click Me");
        Button btn4 = new Button("Click Me");
        Button btn5 = new Button("Click Me");
        Button btn6 = new Button("Click Me");

        GridPane root=new GridPane();

        root.add(btn1,0,0);
        root.add(btn2,1,1);
        root.add(btn3,2,2);
        root.add(btn4,3,3);
        root.add(btn5,4,4);
        root.add(btn6,5,5);
```

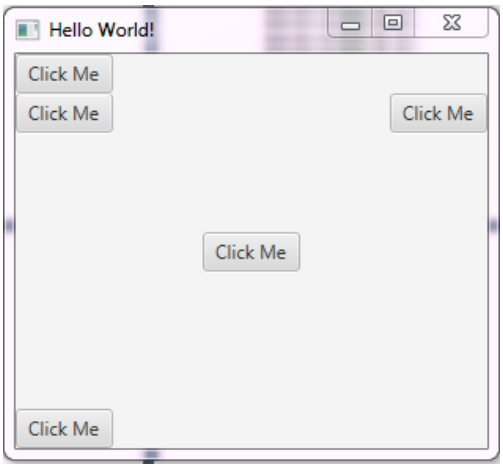
```
        root.setHgap(10);
        root.setVgap(10);
        root.setGridLinesVisible(true);



        Scene scene = new Scene(root, 300, 250);

        primaryStage.setTitle("Hello World!");
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    public static void main(String[] args) {
        launch(args);
    }


}
```



6) Show Button Using BorderPane

```
package layoutexamplehbox;

import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.layout.BorderPane;
import javafx.scene.layout.GridPane;
import javafx.stage.Stage;

public class LayoutExampleHbox extends Application {

    @Override
```

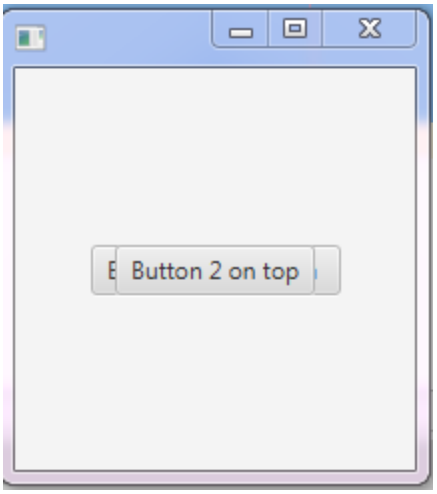
```
    public void start(Stage primaryStage) {
        Button btn1 = new Button("Click Me");
        Button btn2 = new Button("Click Me");
        Button btn3 = new Button("Click Me");
        Button btn4 = new Button("Click Me");
        Button btn5 = new Button("Click Me");

        BorderPane root=new BorderPane();

        root.setBottom(btn1);
        root.setCenter(btn2);
        root.setLeft(btn3);
        root.setRight(btn4);
        root.setTop(btn5);

        Scene scene = new Scene(root, 300, 250);

        primaryStage.setTitle("Hello World!");
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    public static void main(String[] args) {
        launch(args);
    }

}
```



7) Show Button Using StackPane

```
package layoutexamplehbox;

import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Button;
```

```
import javafx.scene.layout.StackPane;
import javafx.stage.Stage;
public class LayoutExampleHbox extends Application {

    @Override
    public void start(Stage primaryStage) throws Exception {
        Button btn1 = new Button("Button 1 on bottom ");
        Button btn2 = new Button("Button 2 on top");
        StackPane root = new StackPane();
        Scene scene = new Scene(root,200,200);
        root.getChildren().addAll(btn1,btn2);
        primaryStage.setScene(scene);
        primaryStage.show();
    }
    public static void main(String[] args) {
        launch(args);
    }

}
```



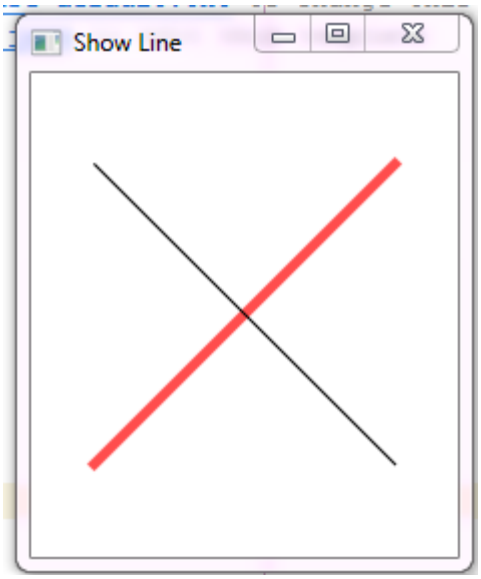
Shape

Shape	Description
Line	In general, Line is the geometrical figure which joins two (X,Y) points on 2D coordinate system. In JavaFX, **javafx.scene.shape.Line** class needs to be instantiated in order to create lines.
Rectangle	In general, Rectangle is the geometrical figure with two pairs of two equal sides and four right angles at their joint. In JavaFX, **javafx.scene.shape.Rectangle** class needs to be instantiated in order to create Rectangles.
Ellipse	In general, ellipse can be defined as a curve with two focal points. The sum of the distances to the focal points are constant from each point of the ellipse. In JavaFX. **javafx.scene.shape.Ellipse** class needs to be instantiated in order to create Ellipse.
Arc	Arc can be defined as the part of the circumference of the circle of ellipse. In JavaFX, **javafx.scene.shape.Arc** class needs to be instantiated in order to create Arcs.
Circle	A circle is the special type of Ellipse having both the focal points at the same location. In JavaFX, Circle can be created by instantiating **javafx.scene.shape.Circle** class.
Polygon	Polygon is a geometrical figure that can be created by joining the multiple Co-planner line segments. In JavaFX, **javafx.scene.shape**. Pollygon class needs to be instantiated in order to create polygon.

8) Draw Line

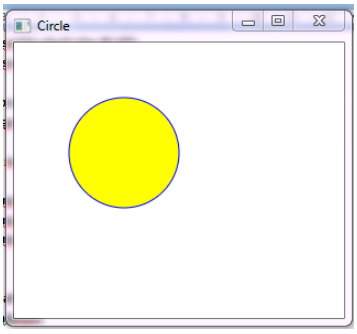
```
package shape_line;

import javafx.scene.paint.Color;
import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.layout.StackPane;
import javafx.scene.shape.Line;
import javafx.stage.Stage;

public class Shape_Line extends Application {

   @Override
   public void start(Stage primaryStage) {
      StackPane pane=new StackPane();
      Line line1=new Line();
      line1.setStartX(10);
      line1.setStartY(10);
      line1.setEndX(160);
      line1.setEndY(160);
      line1.setStrokeWidth(5);
```

```
        line1.setStroke(Color.RED);
        line1.setOpacity(0.7);
        line1.setRotate(90);
         Line line2=new Line(10,10,160,160);
         pane.getChildren().addAll(line1,line2);
         Scene scene=new Scene(pane);

        primaryStage.setTitle("Show Line");
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    public static void main(String[] args) {
        launch(args);
    }

}
```



9) Circle

```
package shape_circle;

import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.layout.Pane;
import javafx.scene.paint.Color;
import javafx.scene.shape.Circle;
import javafx.stage.Stage;


public class Shape_Circle extends Application {

   @Override
   public void start(Stage primaryStage) {
      Circle circle=new Circle();
      circle.setCenterX(100);
      circle.setCenterY(100);
      circle.setRadius(50);
      circle.setStroke(Color.BLUE);
      circle.setFill(Color.YELLOW);

      Pane pane=new Pane();
      pane.getChildren().add(circle);

      Scene scene = new Scene(pane, 300, 250);

      primaryStage.setTitle("Circle");
      primaryStage.setScene(scene);
      primaryStage.show();
   }

   public static void main(String[] args) {
      launch(args);
   }

}
```

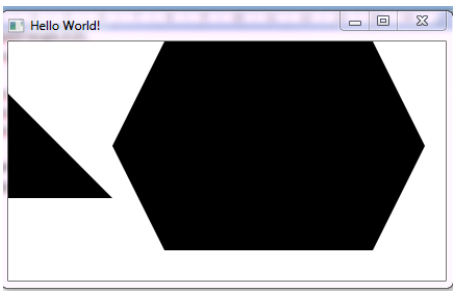


10) **Polygon**

```
import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.layout.GridPane;
import javafx.scene.layout.StackPane;
import javafx.scene.shape.Polygon;
import javafx.stage.Stage;

public class PolygonDemo extends Application {

   @Override
   public void start(Stage primaryStage) {
     Polygon hexagon=new Polygon();
     Polygon triangle=new Polygon();
      hexagon.getPoints().addAll(new Double[]{200.0,50.0,
      400.0,50.0,
      450.0,150.0,
      400.0,250.0,
      200.0,250.0,
      150.0,150.0});
      triangle.getPoints().setAll(200.0,200.0,300.0,300.0,200.0,300.0);
      GridPane root=new GridPane();
       root.add(hexagon,1,0);
       root.add(triangle,0,0);
       Scene scene = new Scene(root, 300, 250);

       primaryStage.setTitle("Hello World!");
       primaryStage.setScene(scene);
       primaryStage.show();
   }
   public static void main(String[] args) {
      launch(args);
   }
}
```

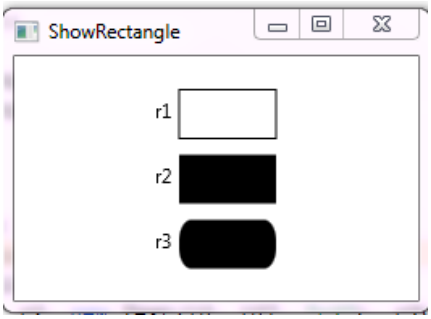
HEXAGONE HAS 6 VERTEX IT MEANS 12VALUES FOR (X,Y)

WE CAN USE SET ALL AND ADD ALL ALSO
HERE TRIANGLE HAS 3 VERTEX SO PASS VALUE 6



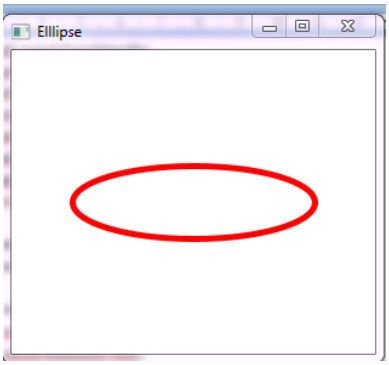
11) Rectangle

```
import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.layout.BorderPane;
import javafx.scene.paint.Color;
import javafx.stage.Stage;
import javafx.scene.text.Text;
import javafx.scene.shape.Rectangle;

public class ShowRectangle extends Application {
 @Override
 public void start(Stage primaryStage) {
  // Create rectangles
  Rectangle r1 = new Rectangle(25, 10, 60, 30);      1-> (X,Y) POINT AND ONE LENGTH,ONE wIDTH
  r1.setStroke(Color.BLACK);      X,Y,LENGTH,WIDTH
  r1.setFill(Color.WHITE);
  Rectangle r2 = new Rectangle(25, 50, 60, 30);
  Rectangle r3 = new Rectangle(25, 90, 60, 30);
  r3.setArcWidth(15);   FOR ROUNDED RECTANGLE IT CUTS ARC AS PERDIMENSION
  r3.setArcHeight(25);

  // Create a group and add nodes to the group
  Group group = new Group();
  group.getChildren().addAll(new Text(10, 27, "r1"), r1,
    new Text(10, 67, "r2"), r2, new Text(10, 107, "r3"), r3);

  // Create a scene and place it in the stage
  Scene scene = new Scene(new BorderPane(group), 250, 150);
  primaryStage.setTitle("ShowRectangle");
  primaryStage.setScene(scene);
  primaryStage.show();
 }
  public static void main(String[] args) {
  launch(args);
 } }
```



12) Ellipse

```
package ellipsedemo;

import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.layout.StackPane;
import javafx.scene.paint.Color;
import javafx.scene.shape.Ellipse;
import javafx.stage.Stage;
public class EllipseDemo extends Application {

   @Override
   public void start(Stage primaryStage) {

      Ellipse ellipse=new Ellipse(150.0f,70.0f,100.0f,30.0f);
      ellipse.setStrokeWidth(5);
      ellipse.setStroke(Color.RED);
      ellipse.setFill(Color.WHITE);
      StackPane root=new StackPane(ellipse);
      Scene scene = new Scene(root, 300, 250);

      primaryStage.setTitle("Elllipse");
      primaryStage.setScene(scene);
      primaryStage.show();
   }

   public static void main(String[] args) {
      launch(args);
   }
}
```

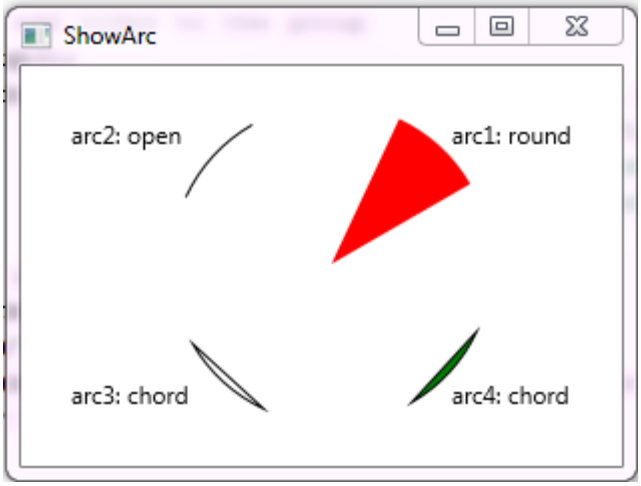


13) Arc

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.Group;
import javafx.scene.layout.BorderPane;
import javafx.scene.paint.Color;
import javafx.stage.Stage;
import javafx.scene.shape.Arc;
import javafx.scene.shape.ArcType;
import javafx.scene.text.Text;

public class ShowArc extends Application {
  @Override // Override the start method in the Application class
  public void start(Stage primaryStage) {
    Arc arc1 = new Arc(150, 100, 80, 80, 30, 35); // Create an arc
    arc1.setFill(Color.RED); // Set fill color
    arc1.setType(ArcType.ROUND); // Set arc type

    Arc arc2 = new Arc(150, 100, 80, 80, 30 + 90, 35);
    arc2.setFill(Color.WHITE);
    arc2.setType(ArcType.OPEN);
    arc2.setStroke(Color.BLACK);

    Arc arc3 = new Arc(150, 100, 80, 80, 30 + 180, 35);
    arc3.setFill(Color.WHITE);
    arc3.setType(ArcType.CHORD);
    arc3.setStroke(Color.BLACK);

    Arc arc4 = new Arc(150, 100, 80, 80, 30 + 270, 35);
    arc4.setFill(Color.GREEN);
    arc4.setType(ArcType.CHORD);
    arc4.setStroke(Color.BLACK);

    // Create a group and add nodes to the group
    Group group = new Group();
    group.getChildren().addAll(new Text(210, 40, "arc1: round"), arc1,
                               new Text(20, 40, "arc2: open"), arc2,
                               new Text(20, 170, "arc3: chord"), arc3,
                               new Text(210, 170, "arc4: chord"), arc4);

    // Create a scene and place it in the stage
    Scene scene = new Scene(new BorderPane(group), 300, 200);
    primaryStage.setTitle("ShowArc"); // Set the stage title
    primaryStage.setScene(scene); // Place the scene in the stage
```

CENTER XY,RAD X,RAD Y,ANGLE,LENGTH

```
    primaryStage.show(); // Display the stage
  }

  public static void main(String[] args) {
    launch(args);
  }
}
```



14) Image

```
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.layout.HBox;
import javafx.scene.layout.Pane;
import javafx.geometry.Insets;
import javafx.stage.Stage;
import javafx.scene.image.Image;
import javafx.scene.image.ImageView;
import javafx.scene.layout.GridPane;
public class ShowImage extends Application {
  @Override // Override the start method in the Application class
  public void start(Stage primaryStage) throws FileNotFoundException {

      GridPane pane = new GridPane();
      pane.setPadding(new Insets(5, 5, 5, 5));
```

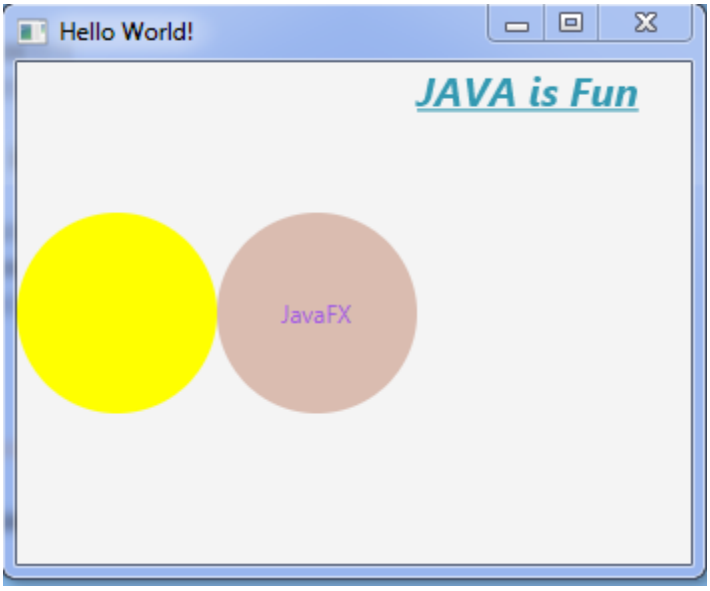
FILE MATHI IMAGE Lavva

```
FileInputStream fin=new FileInputStream("D:\\Core Java\\JavaFX Program\\image\\Capture.png");
      Image image = new Image(fin);

      pane.add(new ImageView(image),0,0);

      ImageView imageView2 = new ImageView(image);
      imageView2.setFitHeight(100);
      imageView2.setFitWidth(100);
      pane.add(imageView2,0,1);

       ImageView imageView3 = new ImageView(image);
       imageView3.setRotate(90);
       pane.add(imageView3,0,2);

       // Create a scene and place it in the stage
       Scene scene = new Scene(pane);
       primaryStage.setTitle("ShowImage"); // Set the stage title
       primaryStage.setScene(scene); // Place the scene in the stage
       primaryStage.show(); // Display the stage
  }
  public static void main(String[] args) {
    launch(args);
   }
}
```

15) Use of Label, Text, Color Methods, Font Class

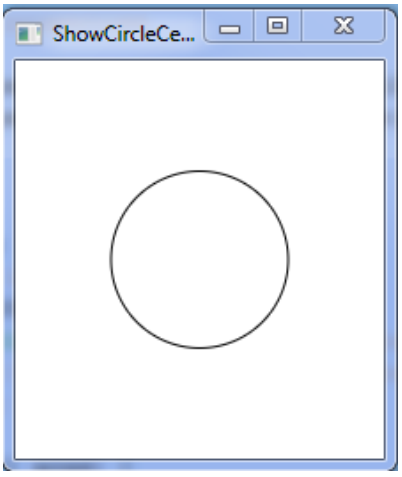
```
package colorlabeltext;

import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.layout.HBox;
import javafx.scene.layout.StackPane;
import javafx.scene.paint.Color;
import javafx.scene.shape.Circle;
import javafx.scene.text.Font;
import javafx.scene.text.FontPosture;
import javafx.scene.text.FontWeight;
import javafx.scene.text.Text;
import javafx.stage.Stage;

public class ColorLabelText extends Application {

   @Override
   public void start(Stage primaryStage) {
      Circle c1=new Circle();
      c1.setRadius(50);
      //Constant Color
      c1.setFill(Color.YELLOW);

      Circle c2=new Circle();
      c2.setRadius(50);
      //rgb COLOR Method
      c2.setFill(Color.rgb(200, 150, 130, 0.6));

      Label l=new Label("JavaFX");
      //hsb COLOR Method
      l.setTextFill(Color.hsb(270, 0.6, 0.9));

      StackPane root=new StackPane();
      root.getChildren().add(c1);

      StackPane root1=new StackPane();
      root1.getChildren().addAll(c2,l);


      Text t=new Text(20,20,"JAVA is Fun");
```

```
        //color Method
        t.setFill(Color.color(0.2, 0.6, 0.7));
        //Font class
        t.setFont(Font.font("Courier",FontWeight.BOLD,FontPosture.ITALIC,20));
        t.setUnderline(true);


        HBox root3=new HBox();
        root3.getChildren().addAll(root,root1,t);

        Scene scene = new Scene(root3, 300, 250);

        primaryStage.setTitle("Hello World!");
        primaryStage.setScene(scene);
        primaryStage.show();
    }
    public static void main(String[] args) {
        launch(args);
    }

}
```



16) Property Bind

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.layout.Pane;
import javafx.scene.paint.Color;
import javafx.scene.shape.Circle;
import javafx.stage.Stage;
public class PropertyBind extends Application {
@Override // Override the start method in the Application class
public void start(Stage primaryStage) {
// Create a pane to hold the circle
Pane pane = new Pane();
// Create a circle and set its properties
Circle circle = new Circle();
circle.centerXProperty().bind(pane.widthProperty().divide(2));
circle.centerYProperty().bind(pane.heightProperty().divide(2));
circle.setRadius(50);
circle.setStroke(Color.BLACK);
circle.setFill(Color.WHITE);
pane.getChildren().add(circle); // Add circle to the pane
// Create a scene and place it in the stage
Scene scene = new Scene(pane, 200, 200);
primaryStage.setTitle("ShowCircleCentered"); // Set the stage title
primaryStage.setScene(scene); // Place the scene in the stage
primaryStage.show(); // Display the stage
}
public static void main(String[] args) {
        System.out.println("launch application");
        Application.launch(args);
        }
}
```



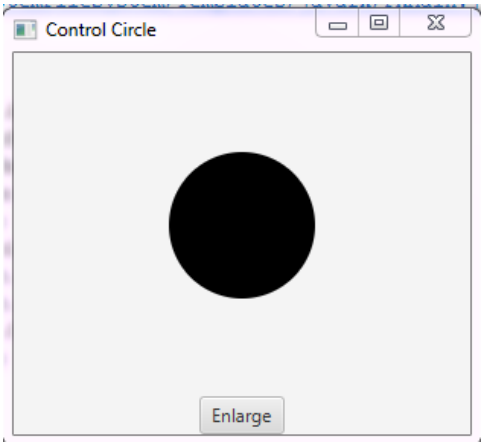
17) Inner Class Handler

```
import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.layout.BorderPane;
import javafx.scene.layout.HBox;
import javafx.scene.shape.Circle;
import javafx.stage.Stage;

public class InnerClassHandler1 extends Application {
   private final Circle circle=new Circle();
   @Override
   public void start(Stage primaryStage)
   {
      HBox hbox=new HBox();
      hbox.setSpacing(10);
      hbox.setAlignment(Pos.CENTER);
      Button btn=new Button("Enlarge");
      hbox.getChildren().add(btn);

     btn.setOnAction(new EnlargeHandler());
      BorderPane borderpane=new BorderPane();
      borderpane.setCenter(circle);
      borderpane.setBottom(hbox);
      BorderPane.setAlignment(hbox,Pos.CENTER);

      Scene scene = new Scene(borderpane, 300, 250);
      primaryStage.setTitle("Control Circle");
      primaryStage.setScene(scene);
      primaryStage.show();
   }
 class EnlargeHandler implements EventHandler<ActionEvent>{
      @Override
      public void handle(ActionEvent e)
      {
         circle.setRadius(circle.getRadius()+2);
      }  }
   public static void main(String[] args) {
      launch(args);
   }
}
```

Inner class

Handle method

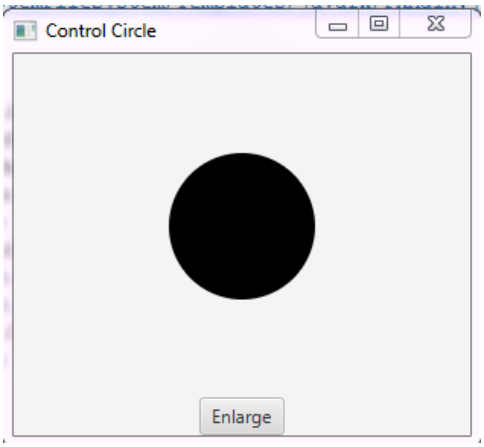
Click on enlarge button



18) Anonymous Inner Class Handler Anonymous class is use for interface

```
import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.layout.BorderPane;
import javafx.scene.layout.HBox;
import javafx.scene.shape.Circle;
import javafx.stage.Stage;

public class InnerClassHandler1 extends Application {

   private final Circle circle=new Circle();
   @Override
   public void start(Stage primaryStage)
   {
      HBox hbox=new HBox();
      hbox.setSpacing(10);
      hbox.setAlignment(Pos.CENTER);
      Button btn=new Button("Enlarge");
      hbox.getChildren().add(btn);

      EventHandler<ActionEvent> handler = new EventHandler<ActionEvent>(){
      @Override
      public void handle(ActionEvent e)
      {
         circle.setRadius(circle.getRadius()+2);
      }
      };

       btn.setOnAction(handler);
       BorderPane borderpane=new BorderPane();
      borderpane.setCenter(circle);
      borderpane.setBottom(hbox);
      BorderPane.setAlignment(hbox,Pos.CENTER);
      Scene scene = new Scene(borderpane, 300, 250);
      primaryStage.setTitle("Control Circle");
      primaryStage.setScene(scene);
      primaryStage.show();
   }
   public static void main(String[] args) {
      launch(args);    } }
```

Click on enlarge button

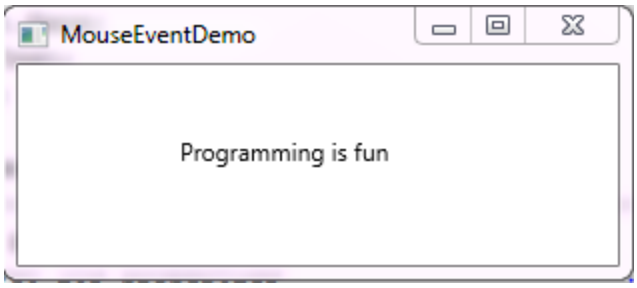


19) Mouse Event only in 3 marks

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.layout.Pane;
import javafx.scene.text.Text;
import javafx.stage.Stage;
public class MouseEventDemo extends Application {
 @Override
 public void start(Stage primaryStage) {
  // Create a pane and set its properties
  Pane pane = new Pane();
  Text text = new Text(20, 20, "Programming is fun");
  pane.getChildren().addAll(text);
  text.setOnMouseDragged(e -> {
   text.setX(e.getX());
   text.setY(e.getY());
  });
  // Create a scene and place it in the stage
  Scene scene = new Scene(pane, 300, 100);
  primaryStage.setTitle("MouseEventDemo"); // Set the stage title
  primaryStage.setScene(scene); // Place the scene in the stage
  primaryStage.show(); // Display the stage
 }
 public static void main(String[] args) {
  launch(args);
 } }
```

Mouse Event

Drag and drop the text using mouse



20) KeyEvent only in 3 marks

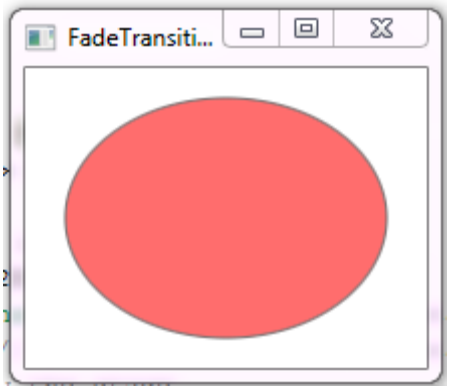
```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.layout.Pane;
import javafx.scene.text.Text;
import javafx.stage.Stage;
public class KeyEventDemo extends Application {
 @Override // Override the start method in the Application class
 public void start(Stage primaryStage) {
  // Create a pane and set its properties
  Pane pane = new Pane();
  Text text = new Text(20, 20, "A");
  pane.getChildren().add(text);
  text.setOnKeyPressed(e -> {      lambda (e->)
            text.setText(e.getText());    });
  Scene scene = new Scene(pane);
  primaryStage.setTitle("KeyEventDemo"); // Set the stage title
  primaryStage.setScene(scene); // Place the scene in the stage
  primaryStage.show(); // Display the stage
  text.requestFocus(); // text is focused to receive key input
 }
 public static void main(String[] args) {
  launch(args);
 }
}
```

Press Key on KeyBoard

21) Fade Transition

```
import javafx.animation.FadeTransition;
import javafx.animation.Timeline;
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.layout.Pane;
import javafx.scene.paint.Color;
import javafx.scene.shape.Ellipse;
import javafx.stage.Stage;
import javafx.util.Duration;

public class FadeTransitionAnimation extends Application {
 @Override // Override the start method in the Application class
 public void start(Stage primaryStage) {
 // Place an ellipse to the pane
 Pane pane = new Pane();
 Ellipse ellipse = new Ellipse(10, 10, 100, 50);
 ellipse.setFill(Color.RED);
 ellipse.setStroke(Color.BLACK);
 ellipse.centerXProperty().bind(pane.widthProperty().divide(2));
 ellipse.centerYProperty().bind(pane.heightProperty().divide(2));
 ellipse.radiusXProperty().bind(
 pane.widthProperty().multiply(0.4));
 ellipse.radiusYProperty().bind(
 pane.heightProperty().multiply(0.4));
 pane.getChildren().add(ellipse);

 // Apply a fade transition to ellipse
 FadeTransition ft =
 new FadeTransition(Duration.millis(3000), ellipse);
 ft.setFromValue(1.0);
 ft.setToValue(0.1);
 ft.setCycleCount(Timeline.INDEFINITE);
 ft.setAutoReverse(true);
 ft.play(); // Start animation

 // Control animation
 ellipse.setOnMousePressed(e -> ft.pause());
 ellipse.setOnMouseReleased(e -> ft.play());

 // Create a scene and place it in the stage
 Scene scene = new Scene(pane, 200, 150);
 primaryStage.setTitle("FadeTransitionDemo"); // Set the stage title
 primaryStage.setScene(scene); // Place the scene in the stage
```

```
primaryStage.show(); // Display the stage

}
public static void main(String[] args) {
                launch(args);
              }
}
```



22) Path Transition

```
import java.io.FileInputStream;
import java.io.FileNotFoundException;
 import javafx.animation.PathTransition;
 import javafx.application.Application;
 import javafx.scene.Scene;
import javafx.scene.image.Image;
 import javafx.scene.image.ImageView;
 import javafx.scene.layout.Pane;
 import javafx.scene.shape.Line;
 import javafx.stage.Stage;
 import javafx.util.Duration;
 public class PathTransitionAnimation extends Application {
 @Override // Override the start method in the Application class
 public void start(Stage primaryStage) throws FileNotFoundException {
 // Create a pane
 Pane pane = new Pane();

 // Add an image view and add it to pane
 FileInputStream fin=new FileInputStream("D:\\Core Java\\JavaFX
Program\\image\\Capture.png");
 Image image=new Image(fin);
 ImageView imageView = new ImageView(image);

 pane.getChildren().add(imageView);

 // Create a path transition
 PathTransition pt = new PathTransition(Duration.millis(10000),
 new Line(300, 500, 300, 100), imageView);
 pt.setCycleCount(5);
 pt.play(); // Start animation

 // Create a scene and place it in the stage
 Scene scene = new Scene(pane, 250, 200);
 primaryStage.setTitle("FlagRisingAnimation"); // Set the stage title
 primaryStage.setScene(scene); // Place the scene in the stage
 primaryStage.show(); // Display the stage
 }
 public static void main(String[] args) {
              launch(args);
            }
 }
```

23) Loan Calculator

EMI= p*r*(1+r)n /((1+r)n -1)
P = Principal loan amount ; N = Loan tenure in months ; R = Monthly interest rate.
TOTAL AMOUNT= EMI * n*12

```
import javafx.application.Application;
import javafx.event.*;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.layout.*;
import javafx.stage.Stage;
import javafx.geometry.*;
public class EMI_Calculator extends Application {
private TextField tfAnnualInterestRate = new TextField();
private TextField tfNumberOfYears = new TextField();
private TextField tfLoanAmount = new TextField();
private TextField tfMonthlyPayment = new TextField();
private TextField tfTotalPayment = new TextField();
private Button btCalculate = new Button("Calculate");
   @Override
   public void start(Stage primaryStage) {
GridPane gridPane = new GridPane();
gridPane.setStyle("-fx-border: 2px solid; -fx-border-color: red;");
TextField tfAnnualInterestRate = new TextField();
TextField tfNumberOfYears = new TextField();
TextField tfLoanAmount = new TextField();
TextField tfMonthlyPayment = new TextField();
TextField tfTotalPayment = new TextField();
Button btCalculate = new Button("Calculate");
gridPane.setAlignment(Pos.CENTER);
gridPane.setHgap(15);
gridPane.setVgap(15);
gridPane.add(new Label("Annual Interest Rate:"), 0, 0);
gridPane.add(tfAnnualInterestRate, 1, 0);
gridPane.add(new Label("Number of Years:"), 0, 1);
gridPane.add(tfNumberOfYears, 1, 1);
gridPane.add(new Label("Loan Amount:"), 0, 2);
gridPane.add(tfLoanAmount, 1, 2);
gridPane.add(new Label("Monthly Payment:"), 0, 3);
gridPane.add(tfMonthlyPayment, 1, 3);
gridPane.add(new Label("Total Payment:"), 0, 4);
gridPane.add(tfTotalPayment, 1, 4);
gridPane.add(btCalculate, 1, 5);
```

```
btCalculate.setOnAction(e->{
        double p = Double.parseDouble(tfLoanAmount.getText());
        double r = Double.parseDouble(tfAnnualInterestRate.getText()) / 1200;
        double n = Double.parseDouble(tfNumberOfYears.getText()) * 12;


        double EMI=p*r*(Math.pow((1+r), n))/(Math.pow((1+r),n)-1);

        tfMonthlyPayment.setText(String.format("Rs.%.2f", EMI));
        tfTotalPayment.setText(String.format("Rs.%.2f", EMI * n));
});
       Scene scene = new Scene(gridPane, 400, 250);
       primaryStage.setTitle("Hello World!");
       primaryStage.setScene(scene);
       primaryStage.show();
   }
   public static void main(String[] args) {
       launch(args);
   }

}
```