UTTARANCHAL UNIVERSITY DEHRADUN NAAC GRADE A+

JAVA FX

UNIT-III ADVANCE JAVA

TOPICS TO BE DISCUSSED

- JavaFX Controls and layouts.
- JavaFX Program Structure Using FXML

JAVAFX LAYOUTS

- Layouts are the highest level container classes which specify the UI designs for scene graph objects.
- The layout serves as the main node for all other nodes. JavaFX offers a range of layout panes that accommodate various types of layouts.
- In JavaFX, Layout determines how the components will appear on the stage.
- It essentially arranges the nodes of the scene graph. In JavaFX, there are multiple built-in layout panes such as HBox, VBox, StackPane, FlowBox, AnchorPane, etc.
- Each layout pane has its own class that must be created to use that specific layout.
- All of these categories are part of the **javafx.scene.layout** package. The **javafx.scene.layout.Pane** class serves as the foundation class for all the default layout classes in JavaFX.

LAYOUT CLASSES

Classes	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.

UTTARANCHAL UNIVERSITY DEHRADUN NAAC GRADE A+

STEPS TO CREATE LAYOUT

1. Instantiate the respective layout class, for example, **HBox root = new HBox();**
2. Setting the properties for the layout, for example, **root.setSpacing(20);**
3. Adding nodes to the layout object, for example,

root.getChildren().addAll(<NodeObjects>);

JAVAFX BORDERPANE

- BorderPane positions the nodes on the left, right, center, top, and bottom of the screen.
- The **javafx.scene.layout.BorderPane** class represents it. This class offers different functions such as setRight(), setLeft(), setCenter(), setBottom() and setTop() to assign the position for specific nodes.
- To create the BorderPane layout, we must create an instance of the BorderPane class.

JAVAFX BORDERPANE -CONSTRUCTORS

There are the following constructors in the class.

1. BorderPane() : create the empty layout
2. BorderPane(Node Center) : create the layout with the center node
3. BorderPane(Node Center, Node top, Node right, Node bottom, Node left) : create the layout with all the nodes

JAVAFX BORDERPANE- EXAMPLE

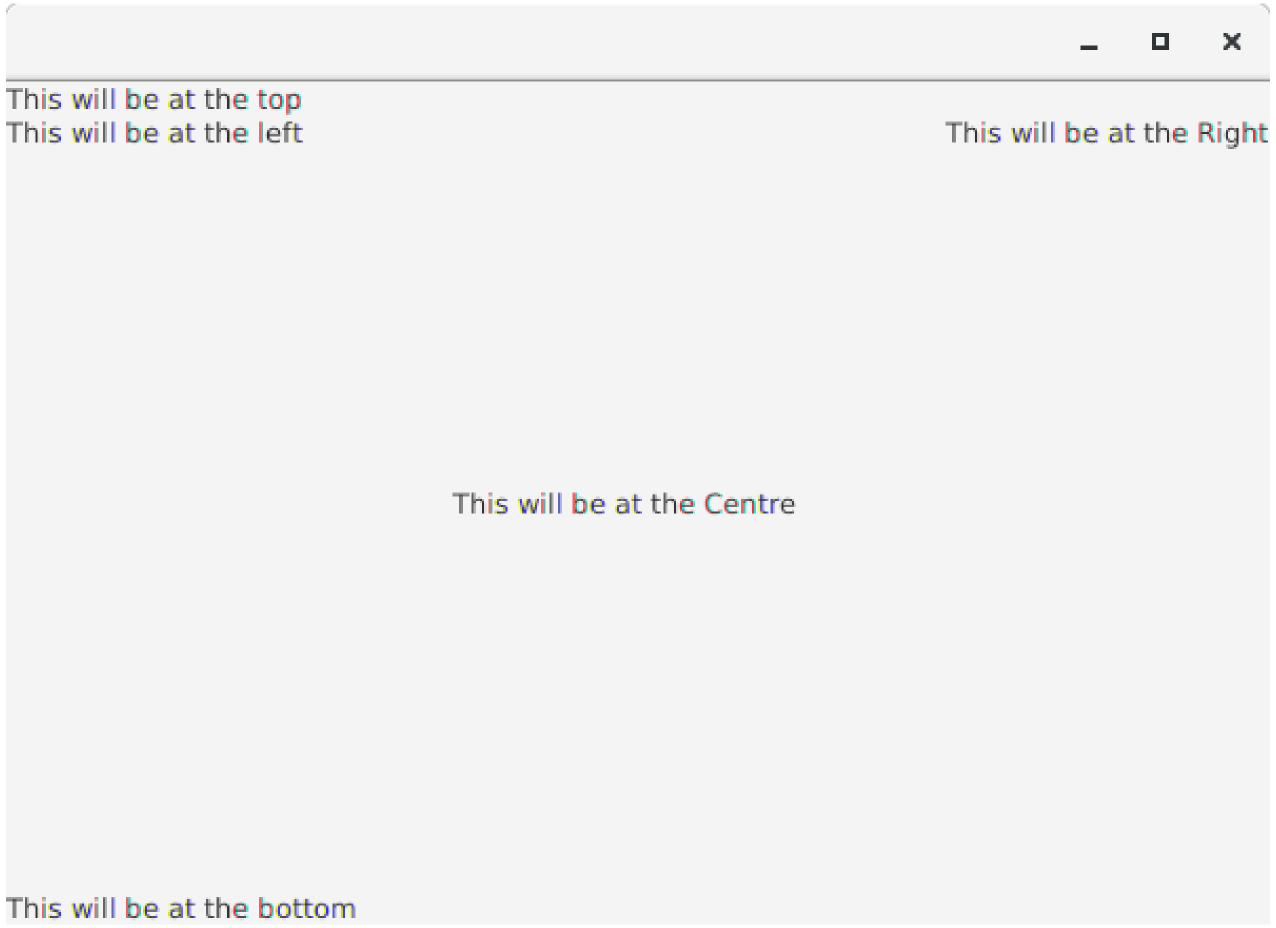
```
public class Label_Test extends Application {
     @Override
   public void start(Stage primaryStage) throws Exception {
       BorderPane BPane = new BorderPane();
       BPane.setTop(new Label("This will be at the top"));
       BPane.setLeft(new Label("This will be at the left"));
       BPane.setRight(new Label("This will be at the Right"));
       BPane.setCenter(new Label("This will be at the Centre"));
```

1.

JAVAFX BORDERPANE- EXAMPLE

```
 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);
  }
}
```

JAVAFX BORDERPANE- EXAMPLE



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



                     This will be at the Centre



This will be at the bottom
```

JAVAFX HBOX

- HBox layout pane arranges the nodes in a single row.
- It is represented by **javafx.scene.layout.HBox** class.

Property	**Description**	**Method**
alignment	This represents the alignment of the nodes.	setAlignment(Double)
fillHeight	This is a boolean property. If you set this property to true the height of the nodes will become equal to the height of the HBox.	setFillHeight(Double)
spacing	This represents the space between the nodes in the HBox. It is of double type.	setSpacing(Double)

UTTARANCHAL UNIVERSITY DEHRADUN NAAC GRADE A+

JAVAFX HBOX- CONSTRUCTORS

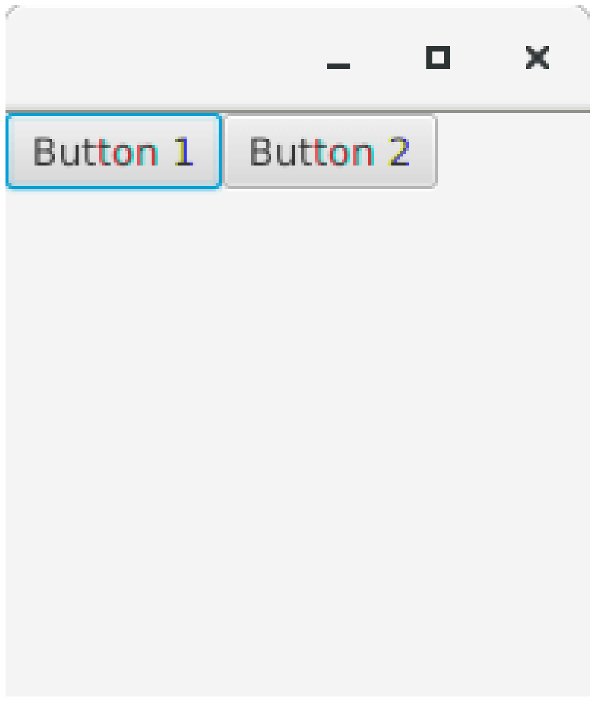
- The HBox class contains two constructors that are given below.

1.new HBox() : create HBox layout with 0 spacing

2.new Hbox(Double spacing) : create HBox layout with a spacing value

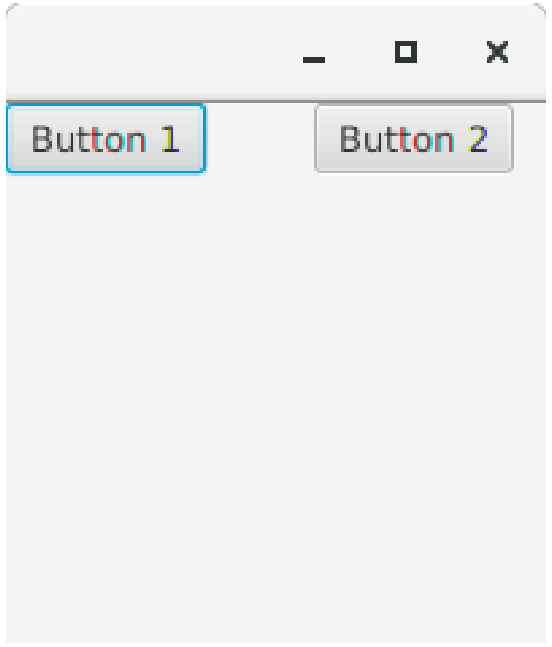
JAVAFX HBOX- EXAMPLE1

```
public class Label_Test extends Application {
public void start(Stage primaryStage) throws Exception {
Button btn1 = new Button("Button 1");
Button btn2 = new Button("Button 2");
HBox root = new HBox();
Scene scene = new Scene(root,200,200);
root.getChildren().addAll(btn1,btn2);
primaryStage.setScene(scene);
primaryStage.show();
}
```



JAVAFX HBOX- EXAMPLE2

```
public class Label_Test extends Application {
 public void start(Stage primaryStage) throws Exception {
Button btn1 = new Button("Button 1");
Button btn2 = new Button("Button 2");
HBox root = new HBox();
Scene scene = new Scene(root,200,200);
root.getChildren().addAll(btn1,btn2);
root.setSpacing(40);
primaryStage.setScene(scene);
primaryStage.show();
}
```



UTTARANCHAL UNIVERSITY DEHRADUN NAAC GRADE A+

JAVAFX VBOX

- Vbox Layout Pane arranges the nodes in a single vertical column.
- It is represented by **javafx.scene.layout.VBox** class which provides all the methods to deal with the styling and the distance among the nodes.

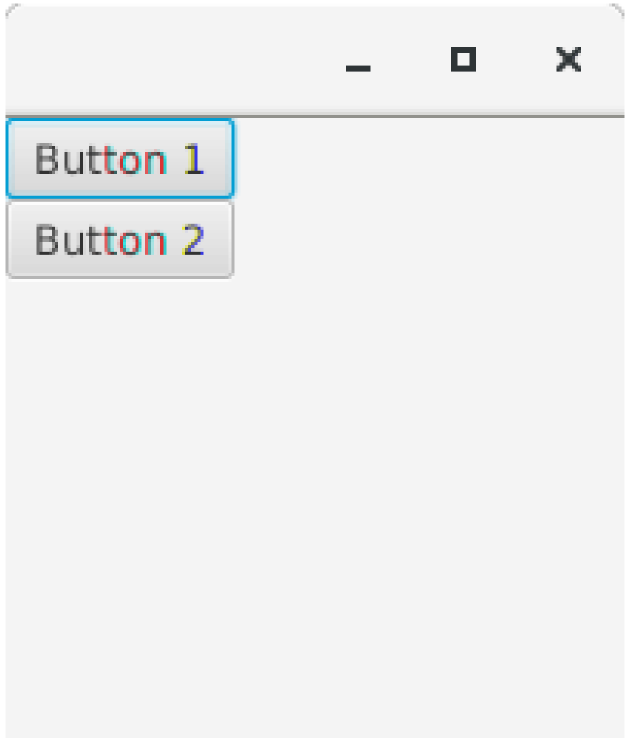
Property	Description	Setter Methods
Alignment	This property is for the alignment of the nodes.	setAlignement(Double)
FillWidth	This property is of the boolean type. The Width of resizeable nodes can be made equal to the Width of the VBox by setting this property to true.	setFillWidth(boolean)
Spacing	This property is to set some spacing among the nodes of VBox.	setSpacing(Double)

JAVAFX VBOX- CONSTRUCTOR

1. **VBox()** : creates layout with 0 spacing
2. **Vbox(Double spacing)** : creates layout with a spacing value of double type
3. **Vbox(Double spacing, Node? children)** : creates a layout with the specified spacing among the specified child nodes
4. **Vbox(Node? children)** : creates a layout with the specified nodes having 0 spacing among them

JAVAFX VBOX- EXAMPLE

```
public class Label_Test extends Application {
    public void start(Stage primaryStage) throws Exception {
        Button btn1 = new Button("Button 1");
        Button btn2 = new Button("Button 2");
        VBox root = new VBox();
        Scene scene = new Scene(root,200,200);
        root.getChildren().addAll(btn1,btn2);
        primaryStage.setScene(scene);
        primaryStage.show();
    }
```



JAVAFX STACKPANE

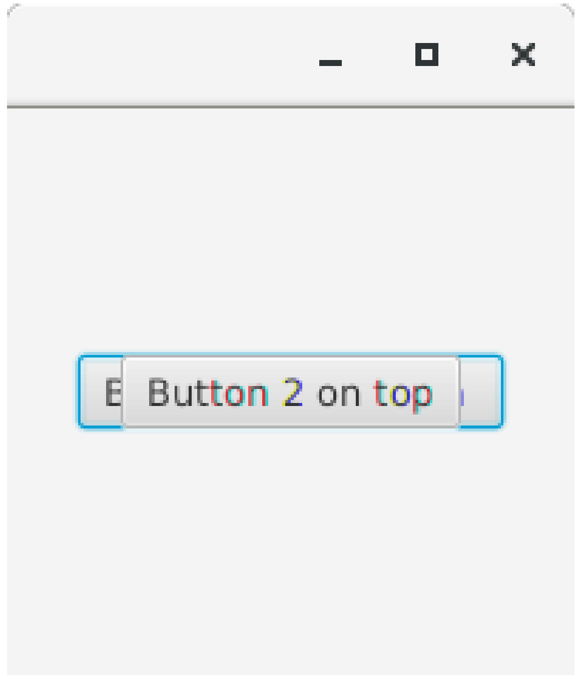
- The StackPane layout pane places all the nodes into a single stack where every new node gets placed on the top of the previous node.
- It is represented by javafx.scene.layout.StackPane class.

- The class contains two constructors that are given below.

1. StackPane()
2. StackPane(Node? Children)

JAVAFX STACKPANE

```
public class Label_Test extends Application {
    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();
   }
```



JAVAFX GRIDPANE

- Grid Pane Layout pane allows us to add the multiple nodes in multiple rows and columns.
- It is seen as a flexible grid of rows and columns where nodes can be placed in any cell of the grid.
- It is represented by **javafx.scence.layout.GridPane** class.
- The class contains only one constructor that is given below.

Public GridPane(): creates a gridpane with 0 hgap/vgap.

JAVAFX GRIDPANE

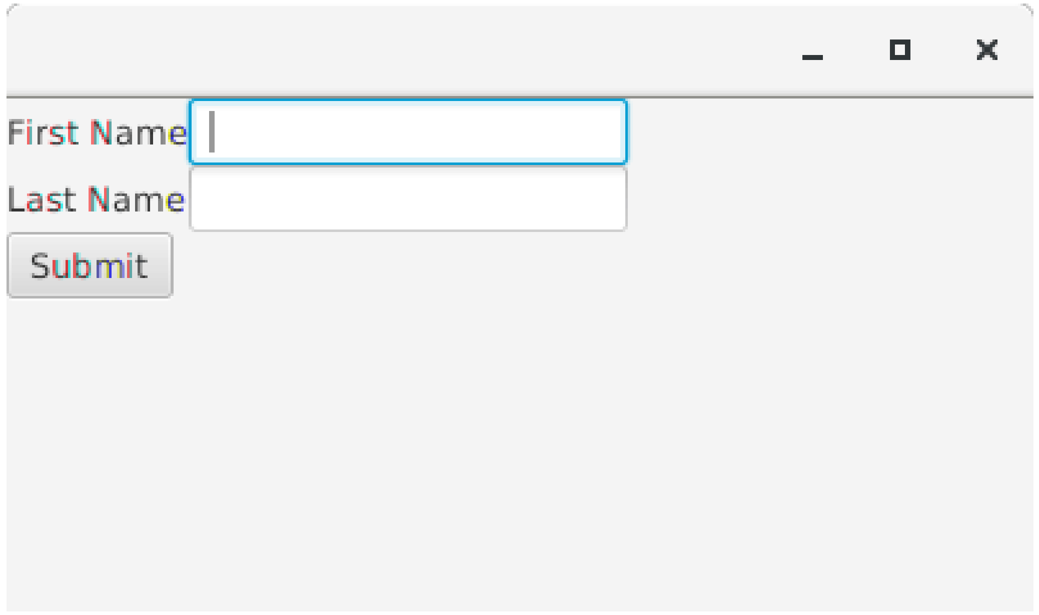
Property	Description	Setter Methods
alignment	Represents the alignment of the grid within the GridPane.	setAlignment(Pos value)
gridLinesVisible	This property is intended for debugging. Lines can be displayed to show the gidpane's rows and columns by setting this property to true.	setGridLinesVisible(Boolean value)
hgap	Horizontal gaps among the columns	setHgap(Double value)
vgap	Vertical gaps among the rows	setVgap(Double value)

EXAMPLE

```
public class Label_Test extends Application {
  public void start(Stage primaryStage) throws Exception {
    Label first_name=new Label("First Name");
    Label last_name=new Label("Last Name");
    TextField tf1=new TextField();
    TextField tf2=new TextField();
    Button Submit=new Button ("Submit");
    GridPane root=new GridPane();
```

EXAMPLE

```
Scene scene = new Scene(root,400,200);
    root.addRow(0, first_name,tf1);
    root.addRow(1, last_name,tf2);
    root.addRow(2, Submit);
    primaryStage.setScene(scene);
    primaryStage.show();
  }
```



JAVAFX FLOWPANE

- Nodes within a FlowPane are arranged in a manner that can wrap along the pane's boundary.
  The nodes are arranged in a row by the horizontal FlowPane and wrapped based on its width.
- The nodes in the vertical FlowPane are arranged in a column and wrapped based on the flow pane's height.
- **The javafx.scene.layout.FlowPane** class represents the FlowPane layout.
- All we have to do is create an instance of this class in order to generate the FlowPane layout.

JAVAFX FLOWPANE

Property	Description	Setter Methods
alignment	The overall alignment of the flowpane's content.	setAlignment(Pos value)
columnHalignment	The horizontal alignment of nodes within the columns.	setColumnHalignment(HPos Value)
hgap	Horizontal gap between the columns.	setHgap(Double value)
orientation	Orientation of the flowpane	setOrientation(Orientation value)
prefWrapLength	The preferred height or width where content should wrap in the horizontal or vertical flowpane.	setPrefWrapLength(double value)
rowValignment	The vertical alignment of the nodes within the rows.	setRowValignment(VPos value)
vgap	The vertical gap among the rows	setVgap(Double value)

JAVAFX FLOWPANE

There are 8 constructors in the class that are given below.

1. FlowPane()
2. FlowPane(Double Hgap, Double Vgap)
3. FlowPane(Double Hgap, Double Vgap, Node? children)
4. FlowPane(Node... Children)
5. FlowPane(Orientation orientation)
6. FlowPane(Orientation orientation, double Hgap, Double Vgap)
7. FlowPane(Orientation orientation, double Hgap, Double Vgap, Node? children)
8. FlowPane(Orientation orientation, Node... Children)

JAVAFX FLOWPANE EXAMPLE

```
public class FlowPaneTest extends Application {
    public void start(Stage primaryStage) {
      primaryStage.setTitle("FlowPane Example");
      FlowPane root = new FlowPane();
      root.setVgap(6);
      root.setHgap(5);
      root.setPrefWrapLength(250);
      root.getChildren().add(new Button("Start"));
```

JAVAFX FLOWPANE EXAMPLE

```
root.getChildren().add(new Button("Stop"));
        root.getChildren().add(new Button("Reset"));
        Scene scene = new Scene(root,300,200);
           primaryStage.setScene(scene);
        primaryStage.show();
    }
```

JAVAFX UI CONTROLS

- The UI design of all desktop applications primarily focuses on UI components, arrangements and interactions.
- The UI elements are those that are displayed to the user for interaction or exchanging information.
- The arrangement of UI elements on the screen is determined by the layout.
- Behaviour refers to the response of the UI element to an event that occurs on it.
- The **javafx.scene.control** package includes all the essential classes for UI elements such as Button, Label, and more.
- Each class corresponds to a particular UI component and includes methods for customizing their appearance.

JAVAFX UI CONTROLS

Control	Description
Label	Label is a component that is used to define a simple text on the screen. Typically, a label is placed with the node, it describes.
Button	Button is a component that controls the function of the application. Button class is used to create a labelled button.
RadioButton	The Radio Button is used to provide various options to the user. The user can only choose one option among all. A radio button is either selected or deselected.
CheckBox	Check Box is used to get the kind of information from the user which contains various choices. User marked the checkbox either on (true) or off(false).
TextField	Text Field is basically used to get the input from the user in the form of text. javafx.scene.control.TextField represents TextField

JAVAFX UI CONTROLS

PasswordField	**PasswordField is used to get the user's password. Whatever is typed in the passwordfield is not shown on the screen to anyone.**
HyperLink	HyperLink are used to refer any of the webpage through your appication. It is represented by the class **javafx.scene.control.HyperLink**
Slider	Slider is used to provide a pane of options to the user in a graphical form where the user needs to move a slider over the range of values to select one of them.
ProgressBar	Progress Bar is used to show the work progress to the user. It is represented by the class **javafx.scene.control.ProgressBar**.
ProgressIndicator	Instead of showing the analogue progress to the user, it shows the digital progress so that the user may know the amount of work done in percentage.

UTTARANCHAL UNIVERSITY DEHRADUN NAAC GRADE A+

JAVAFX UI CONTROLS

ScrollBar	**JavaFX Scroll Bar is used to provide a scroll bar to the user so that the user can scroll down the application pages.**
Menu	JavaFX provides a Menu class to implement menus. Menu is the main component of any application.
ToolTip	JavaFX ToolTip is used to provide hint to the user about any component. It is mainly used to provide hints about the text fields or password fields being used in the application.

JAVAFX UI CONTROLS-RADIO BUTTON

- The Radio Button is used to provide various options to the user. The user can only choose one option among all.
- A radio button is either selected or deselected. It can be used in a scenario of multiple choice questions in the quiz where only one option needs to be chosen by the student.

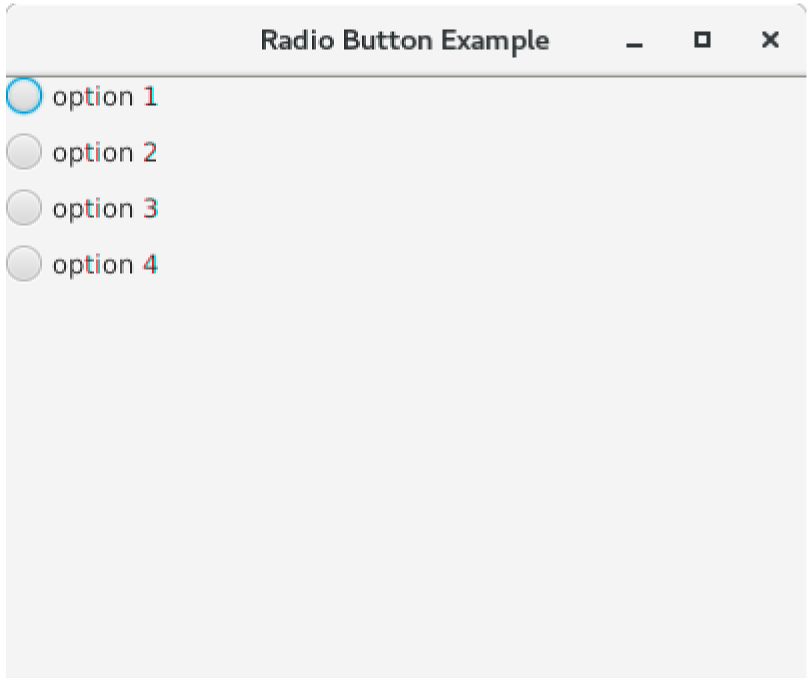
```
public class RadioButtonTest extends Application {
public void start(Stage primaryStage) throws Exception {
    ToggleGroup group = new ToggleGroup();
    RadioButton button1 = new RadioButton("option 1");
    RadioButton button2 = new RadioButton("option 2");
    RadioButton button3 = new RadioButton("option 3");
    RadioButton button4 = new RadioButton("option 4");
```

EXAMPLE

```
button1.setToggleGroup(group);
  button2.setToggleGroup(group);
  button3.setToggleGroup(group);
  button4.setToggleGroup(group);
  VBox root=new VBox();
  root.setSpacing(10);
  root.getChildren().addAll(button1,button2,button3,button4);
```

EXAMPLE

```
 Scene scene=new Scene(root,400,300);
    primaryStage.setScene(scene);
    primaryStage.setTitle("Radio Button Example");
    primaryStage.show();
} }
```

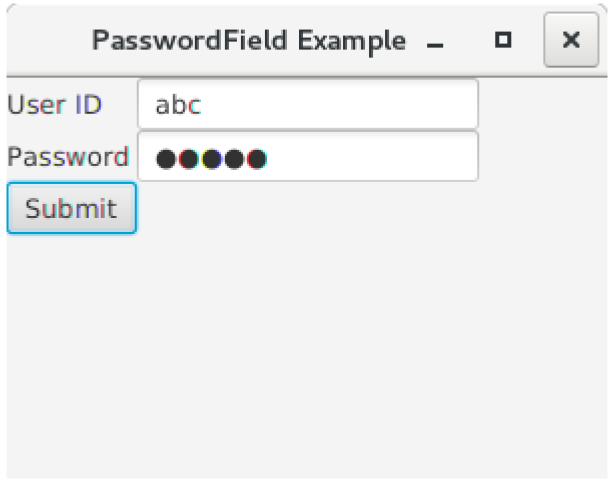

Radio Button Example

TEXTBOX, LABEL AND PASSWORD TYPE -EXAMPLE

```
public void start(Stage primaryStage) throws Exception {
    // TODO Auto-generated method stub
    Label user_id=new Label("User ID");
    Label password = new Label("Password");
    TextField tf=new TextField();
    PasswordField pf=new PasswordField();
    pf.setPromptText("Enter Password");
    Button b = new Button("Submit");
```

TEXTBOX, LABEL AND PASSWORD TYPE -EXAMPLE

```
GridPane root = new GridPane();
    root.addRow(0, user_id, tf);
    root.addRow(1, password, pf);
    root.addRow(5, b);
    Scene scene=new Scene(root,300,200);
    primaryStage.setScene(scene);
    primaryStage.setTitle("PasswordField Example");
    primaryStage.show();
}
}
```


PasswordField Example

User ID abc

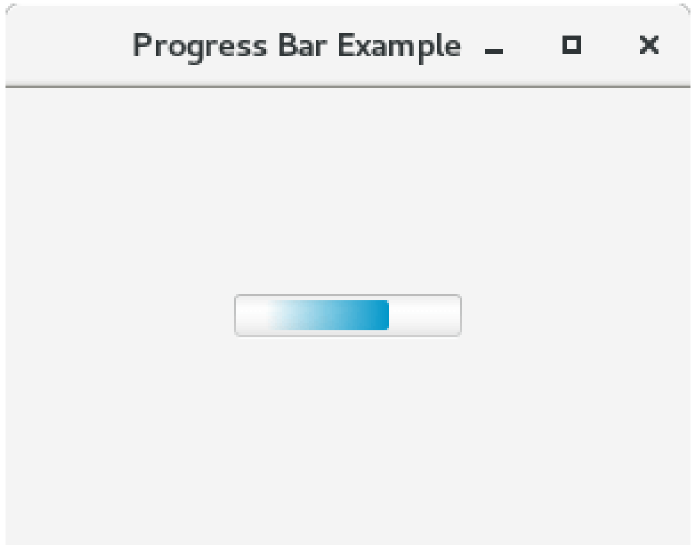
Password ●●●●●

HYPERLINK EXAMPLE

```
public void start(Stage primaryStage) throws Exception {
      Hyperlink hp = new Hyperlink("http://www. uttaranchaluniversity.com");
    StackPane root = new StackPane();
    hp.setOnAction(e -> System.out.println("Link Clicked"));
    root.getChildren().add(hp);
    Scene scene=new Scene(root,400,300);
    primaryStage.setScene(scene);
    primaryStage.setTitle("Hyperlink Example");
    primaryStage.show();
} }
```

PROGRESS BAR

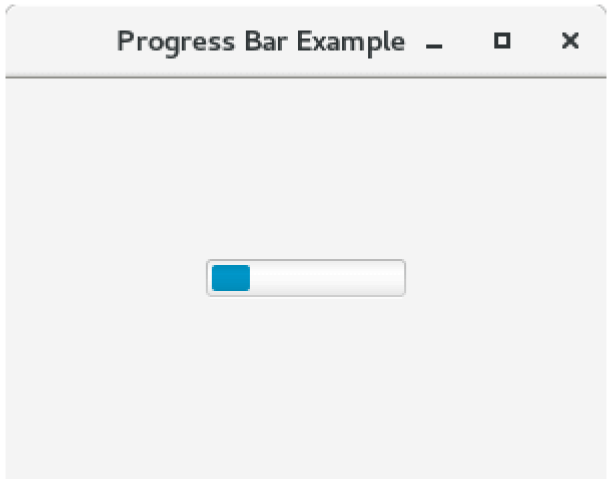
```
public class ProgressBarTest extends Application {
public void start(Stage primaryStage) throws Exception {
     StackPane root = new StackPane();
    ProgressBar progress = new ProgressBar();
    root.getChildren().add(progress);
    Scene scene = new Scene(root,300,200);
    primaryStage.setScene(scene);
    primaryStage.setTitle("Progress Bar Example");
    primaryStage.show();
} }
```



PROGRESS BAR

- Using setProgress() Method

```
ProgressBar p2 = new ProgressBar();
p2.setProgress(0.25f);
```

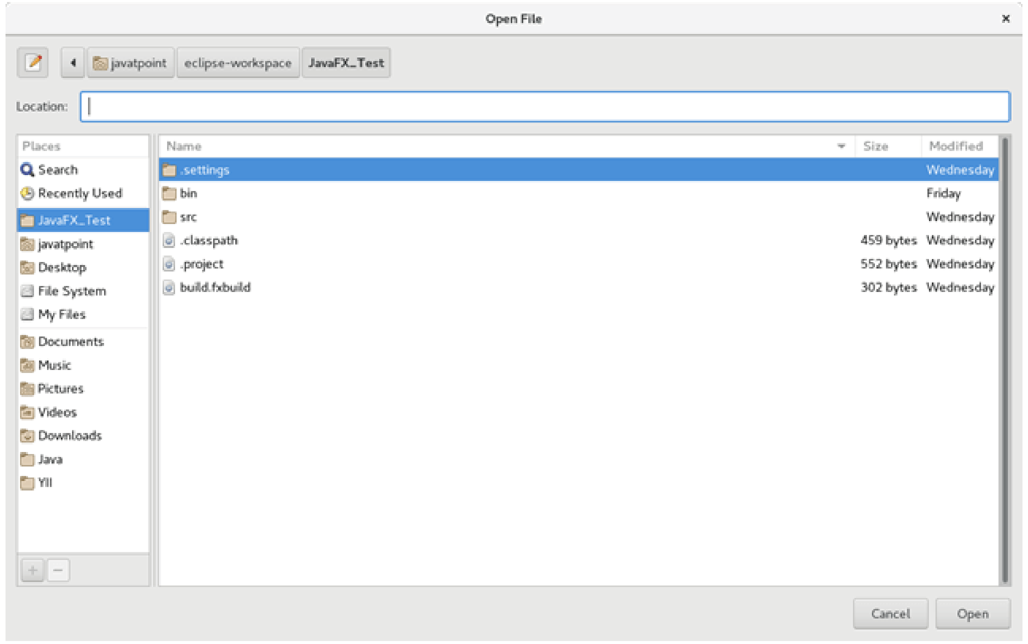


FILE CHOOSER

```
public class FileChooserExample extends Application{
   public void start(Stage primaryStage) throws Exception {
     FileChooser file = new FileChooser();
     file.setTitle("Open File");
     file.showOpenDialog(primaryStage);
       HBox root = new HBox();
       root.setSpacing(20);
```

FILE CHOOSER

```
Scene scene = new Scene(root,350,100);
        primaryStage.setScene(scene);
        primaryStage.setTitle("FileChooser Example");
        primaryStage.show();
 }    }
```



Name	Size	Modified
.settings		Wednesday
bin		Friday
src		Wednesday
.classpath	459 bytes	Wednesday
.project	552 bytes	Wednesday
build.fxbuild	302 bytes	Wednesday