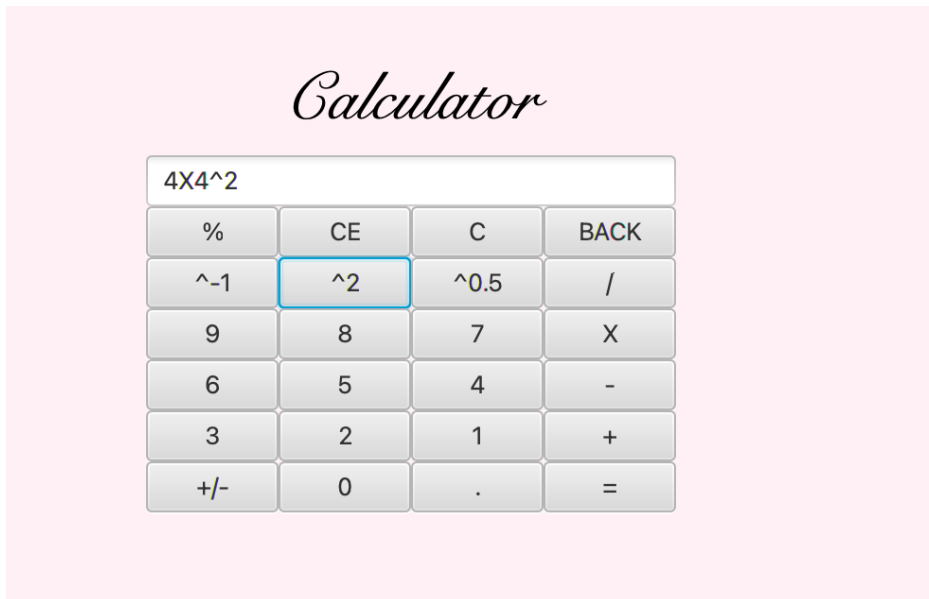


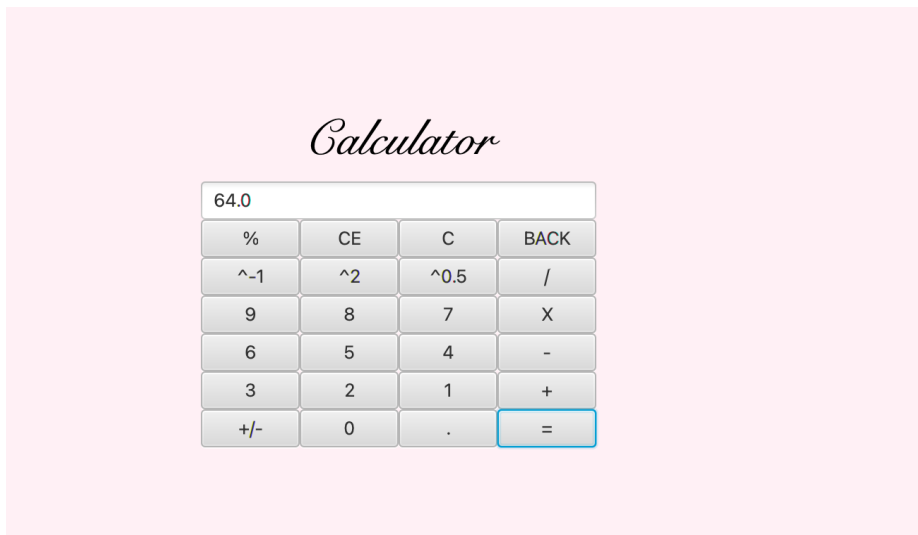
Lab Exercise 11 CSE1007
-AISHWARYA S 19BCE1709

1.

EXAMPLE 1:



RESULT:



EXAMPLE 2:

Calculator

5X5^-1

%	CE	C	BACK
<div>^-1</div>	^2	^0.5	/
9	8	7	X
6	5	4	-
3	2	1	+
+/-	0	.	=

RESULT:

Calculator

1.0

%	CE	C	BACK
^-1	^2	^0.5	/
9	8	7	X
6	5	4	-
3	2	1	+
+/-	0	.	<div>=</div>

EXAMPLE 3:

Calculator

5.5+1.1^2

%	CE	C	BACK
^-1	^2	^0.5	/
9	8	7	X
6	5	4	-
3	2	1	+
+/-	0	.	=

RESULT:

Calculator

6.71

%	CE	C	BACK
^-1	^2	^0.5	/
9	8	7	X
6	5	4	-
3	2	1	+
+/-	0	.	=

EXAMPLE 4:

Calculator

-4X4

%	CE	C	BACK
⁻¹	²	^{0.5}	/
9	8	7	X
6	5	4	-
3	2	1	+
+/-	0	.	=

RESULT:

Calculator

-16.0

%	CE	C	BACK
⁻¹	²	^{0.5}	/
9	8	7	X
6	5	4	-
3	2	1	+
+/-	0	.	=

EXAMPLE 5:

Calculator

8^2%5

%	CE	C	BACK
^-1	^2	^0.5	/
9	8	7	X
6	5	4	-
3	2	1	+
+/-	0	.	=

RESULT:

Calculator

4.0

%	CE	C	BACK
^-1	^2	^0.5	/
9	8	7	X
6	5	4	-
3	2	1	+
+/-	0	.	=

EXAMPLE 6:

Calculator

8/8/2

%	CE	C	BACK
⁻¹	²	^{0.5}	/
9	8	7	X
6	5	4	-
3	2	1	+
+/-	0	.	=

RESULT:

Calculator

2.0

%	CE	C	BACK
⁻¹	²	^{0.5}	/
9	8	7	X
6	5	4	-
3	2	1	+
+/-	0	.	=

EXAMPLE 7:

Calculator

888			
%	CE	C	BACK
⁻¹	²	^{0.5}	/
9	8	7	X
6	5	4	-
3	2	1	+
+/-	0	.	=

PRESSING BACK:

Calculator

88			
%	CE	C	BACK
⁻¹	²	^{0.5}	/
9	8	7	X
6	5	4	-
3	2	1	+
+/-	0	.	=

PRESSING C(CLEAR)

Calculator

%	CE	C	BACK
⁻¹	²	^{0.5}	/
9	8	7	X
6	5	4	-
3	2	1	+
+/-	0	.	=

INVALID INPUTS:
EXAMPLE 1:

Calculator

^2

%	CE	C	BACK
^-1	^2	^0.5	/
9	8	7	X
6	5	4	-
3	2	1	+
+/-	0	.	=

RESULT:

Calculator

invalid

%	CE	C	BACK
^-1	^2	^0.5	/
9	8	7	X
6	5	4	-
3	2	1	+
+/-	0	.	=

EXAMPLE 2:

Calculator

1^0.5/0

%	CE	C	BACK
^-1	^2	^0.5	/
9	8	7	X
6	5	4	-
3	2	1	+
+/-	0	.	=

Calculator

invalid

%	CE	C	BACK
^-1	^2	^0.5	/
9	8	7	X
6	5	4	-
3	2	1	+
+/-	0	.	=

CODE:

```
package jlabs;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Stack;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

import javafx.application.Application;

import javafx.geometry.Insets;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.TextField;

import javafx.scene.layout.Background;
import javafx.scene.layout.BackgroundFill;

import javafx.scene.layout.CornerRadii;
import javafx.scene.layout.GridPane;
import javafx.scene.layout.VBox;
import javafx.scene.text.Text;
import javafx.stage.Stage;
import javafx.scene.paint.Color;

public class lab11 extends Application {
    static Stack<String> stack = new Stack<String>();
    static String string = "";
    static ArrayList<String> al = new ArrayList<String>();
    static ArrayList<String> val = new ArrayList<String>();
    static ArrayList<Double> res = new ArrayList<Double>();

    static int flag = 0;
    static int nprev = -1;
    static int len = 0;
    HashMap<String, Integer> hp = new HashMap<String, Integer>();

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

    String calculate()
    {
        int n = al.size() - 1;
        String op;
        Double num;
        Double n1, n2;
        while(n >= 0)
        {
```

```

        val.add(al.get(n));
        n--;
    }
    System.out.print("STACK");
    for(int k=0;k<val.size();k++)
        System.out.print(val.get(k));
    System.out.print("\n");

    for(int i=0;i<val.size();i++)
    {
        op=val.get(i);

        if(hp.containsKey(op))
        { try
          {
              if(op.equals("+"))
              {
                  int k=res.size();
                  n1=res.get(k-1);

                  n2=res.get(k-2);
                  res.remove(k-1);
                  res.remove(k-2);
                  n1=n1+n2;
                  res.add(n1);

              }
              else if(op.equals("-"))
              {
                  int k=res.size();
                  n1=res.get(k-1);

                  n2=res.get(k-2);
                  res.remove(k-1);
                  res.remove(k-2);
                  n1=n2-n1;
                  res.add(n1);

              }

              else if(op.equals("X"))
              {
                  int k=res.size();
                  n1=res.get(k-1);
                  n2=res.get(k-2);

                  res.remove(k-1);

                  res.remove(k-2);
                  n1=n2*n1;
                  res.add(n1);

              }
              else if(op.equals("/"))
              {
                  int k=res.size();
                  n1=res.get(k-1);

```

```

        n2=res.get(k-2);
        res.remove(k-1);
        res.remove(k-2);
        if(n1==0)
            return "invalid";
        n1=n2/n1;
        res.add(n1);
    }
    else if(op.equals("%"))
    {
        int k=res.size();
        n1=res.get(k-1);

        n2=res.get(k-2);
        res.remove(k-1);
        res.remove(k-2);
        if(n1==0)
            return "invalid";
        n1=n2%n1;
        res.add(n1);
    }

    else if(op.equals("^"))
    { int k=res.size();
      n1=res.get(k-1);
      n2=res.get(k-2);
      res.remove(k-1);
      res.remove(k-2);

      n1=Math.pow(n2, n1);
      res.add(n1);
    }

```

```

    }catch(Exception e)
    {

```

```

        System.out.print(e);
        return "invalid";

```

```

    }

```

```

    }

```

```

    else
    {

```

```

        try
        { if(!op.isEmpty())
            {num=Double.parseDouble(op);
              res.add(num);}

```

```

    }
    catch(Exception e)
    {

        System.out.print(e);
        return "invalid";

    }

}

return "valid";
}
void add(String s)
{
    Pattern p = Pattern.compile("[0-9]*[.][0-9]*$");
    Matcher m = p.matcher(s);
    boolean b = m.matches();
    int n=al.size()-1;
    int f=0;

    System.out.print("\n");

```

```

    if(hp.containsKey(s))

    {

        int op=hp.get(s);

        String k="";
        if(n<0)
            al.add(s);

        else
        {
            while(n>=0)
            {
                if(hp.get(al.get(n))==op)
                {
                    if(s.equals(al.get(n)))
                    {f++;
                     al.add(s);
                    }

                    else
                    {
                        k=al.get(n);
                        al.remove(n);

```

```

        }
    }
    else if(hp.get(al.get(n))>op)
    {
        k=al.get(n);
        al.remove(n);

    }
    else if(hp.get(al.get(n))<op)
    { al.add(s);
      f++;
      break;

    }

    val.add(k);

    n--;
}
if(f==0)
    al.add(s);
}
}

```

else

```

    { if(!s.isEmpty())
      val.add(s);

    }

}

void parse(String s )
{ int i=0;
  String k="";
  String k1="";

  while(i<s.length())
  { k1="" + s.charAt(i);

    if(hp.containsKey(k1))
    {

        if(k1.equals("-"))
        {
            if(i==0)

```

```

        flag=-1;
    else if(hp.containsKey(""+s.charAt(i-1)))
    {
        flag=-1;
    }

    else
    {
        if(!k.isEmpty())
            add(k);
        add(k1);

        k="";
    }
}

else

{
    if(!k.isEmpty())
        add(k);
        add(k1);
        k="";
}

}

else
{
    if(flag==1)
    {
        k=k+s.charAt(i-1)+k1;
        flag=0;
    }

    else
        k=k+k1;
}
i++;
}
if(!k.isEmpty())
    add(k);
}
}

```

```

@Override
public void start(Stage primaryStage) throws Exception {

```

```

hp.put("+",0);
hp.put("-",0);
hp.put("X",1);
hp.put("/",1);
hp.put("%",1);
hp.put("^",2);

```

```

primaryStage.setTitle("19bCE1709 AISHWARYA S");
String[] charstr= new String[24];
String[] s= {"%", "CE", "C", "BACK", "^-1", "^2", "^0.5", "/"};

```

```

int pos=0,i=9;
for(String st:s)
{ charstr[pos]= new String(st);
pos++;
}

```

```

while(i>0)
{
    if(pos==11)
        charstr[pos]=new String("X");
    else if(pos==15)
        charstr[pos]=new String("-");

    else
        {charstr[pos]=new String(Integer.toString(i));

        i--;
        }
    pos++;
}
charstr[pos]=new String("+");
pos++;
charstr[pos]=new String("/-");
pos++;
charstr[pos]=new String("0");
pos++;
charstr[pos]=new String(".");
pos++;
charstr[pos]=new String("=");

```

```

Button[] barr= new Button[24];
GridPane gridPane = new GridPane();

```

```

gridPane.setAlignment(Pos.CENTER);

```



```

        TextField tf = new TextField();

tf.setPrefWidth(280);

gridPane.add(tf,0,0,4,1);
        int row=1, col=0;

        for(i=0;i<24;i++)
        {
            barr[i]= new Button(charstr[i]);
            int j=i;
            barr[i].setOnAction(event -> {

                if(barr[j].getText().equals("="))
                {
                    parse(tf.getText());
                    String k=calculate();
                    System.out.print("\n");
                    for(int m=0;m<val.size();m++)
                        System.out.print(" "+val.get(m));
                    System.out.print("\n");

                    if(k.equals("invalid"))
                        {tf.clear();
                    tf.setText(k);
                    val.clear();
                        }

                    else
                    {

                        k= Double.toString(res.get(0));
                        tf.clear();
                        tf.setText(k);

                        val.clear();

                        val.add(k);
                    }

                    al.clear();
                    res.clear();

                }

            else
            {
                if((barr[j].getText()).equals("C"))
                {
                    al.clear();
                    val.clear();
                    res.clear();
                    nprev=-1;
                    flag=0;

                    tf.clear();
                }
                else if((barr[j].getText()).equals("BACK"))

```

```

        { int m=tf.getText().length()-1;
if(m>=0)
        { String k= tf.getText().substring(0,m);
            tf.setText(k);
        }
        }

else
    {

        String k= tf.getText()+barr[j].getText();
        tf.setText(k);

    }

}

});
barr[i].setPrefWidth(70);
gridPane.add(barr[i],col%4,row);

col=col+1;
row=(col/4)+1;

}

String style = getClass().getResource("application.css").toExternalForm();

    BackgroundFill background_fill = new BackgroundFill(Color.LAVENDERBLUSH,
    CornerRadii.EMPTY, Insets.EMPTY);
    Background background = new Background(background_fill);
    VBox vbox = new VBox();
    vbox.setPadding(new Insets(10));
    vbox.setSpacing(8);
    Text title = new Text("Calculator");
    title.setId("title");
    vbox.getChildren().add(title);
    vbox.getChildren().add(gridPane);

vbox.setId("pane");
vbox.setAlignment(Pos.CENTER);

    Scene scene = new Scene(vbox,200,300,Color.LAVENDERBLUSH);
    scene.getStylesheets().addAll(style);
    primaryStage.setScene(scene);
    primaryStage.show();

}

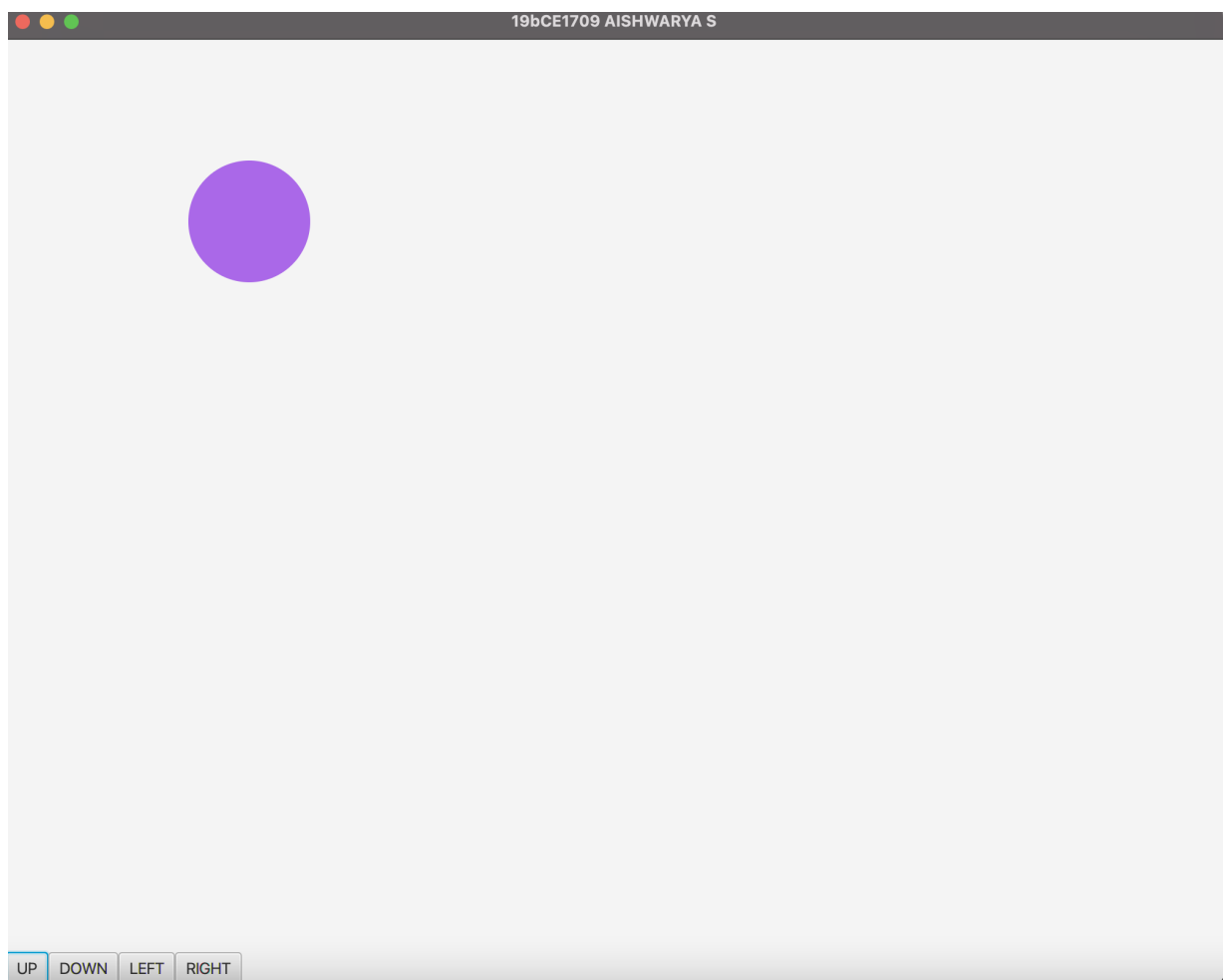
}

```

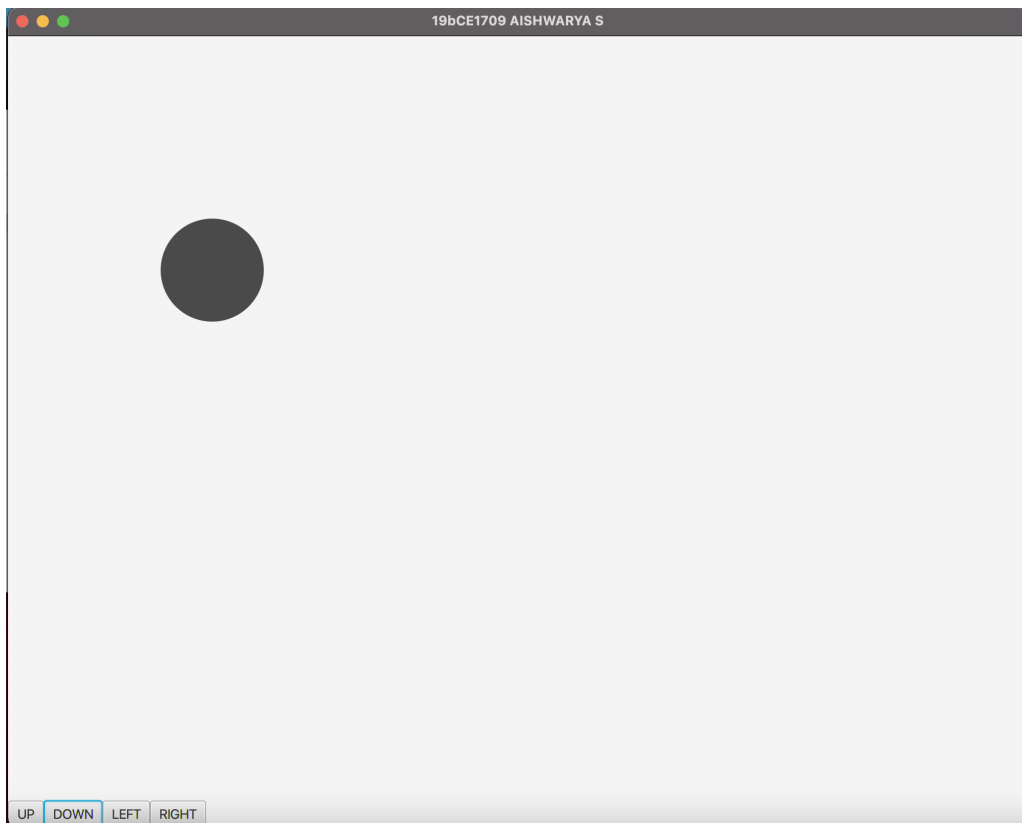
CSS:

```
#pane
{
  -fx-background-color: lilac;
}
#title
{
  -fx-font: 35px 'Snell Roundhand', cursive;
}
```

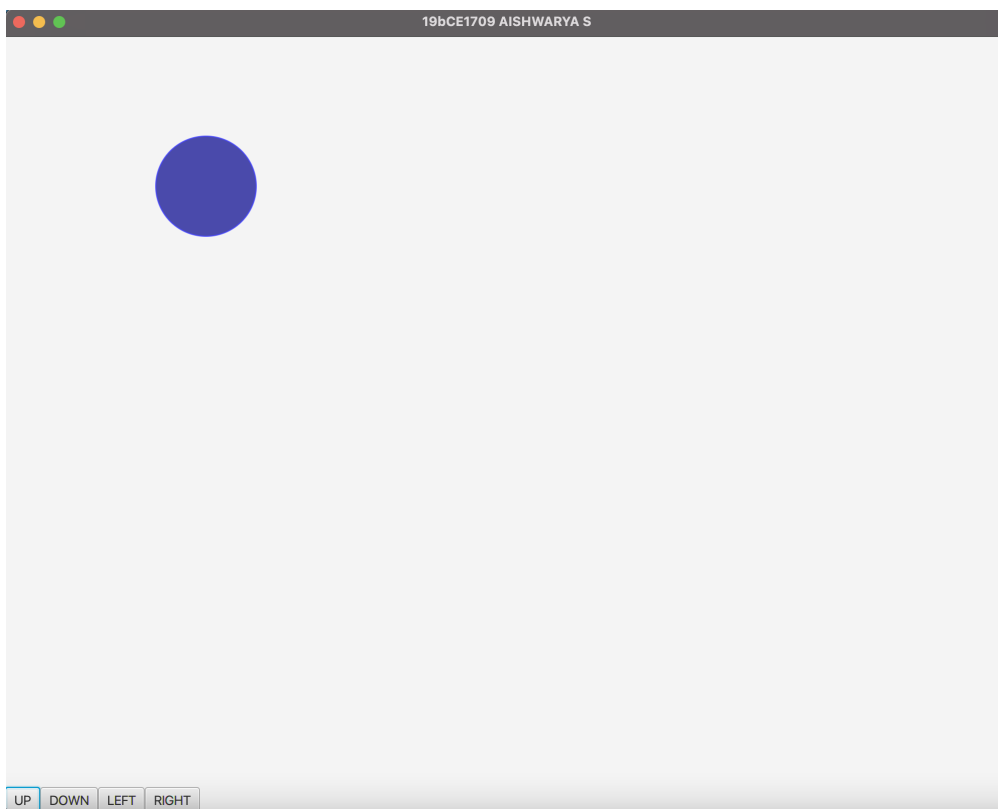
2.



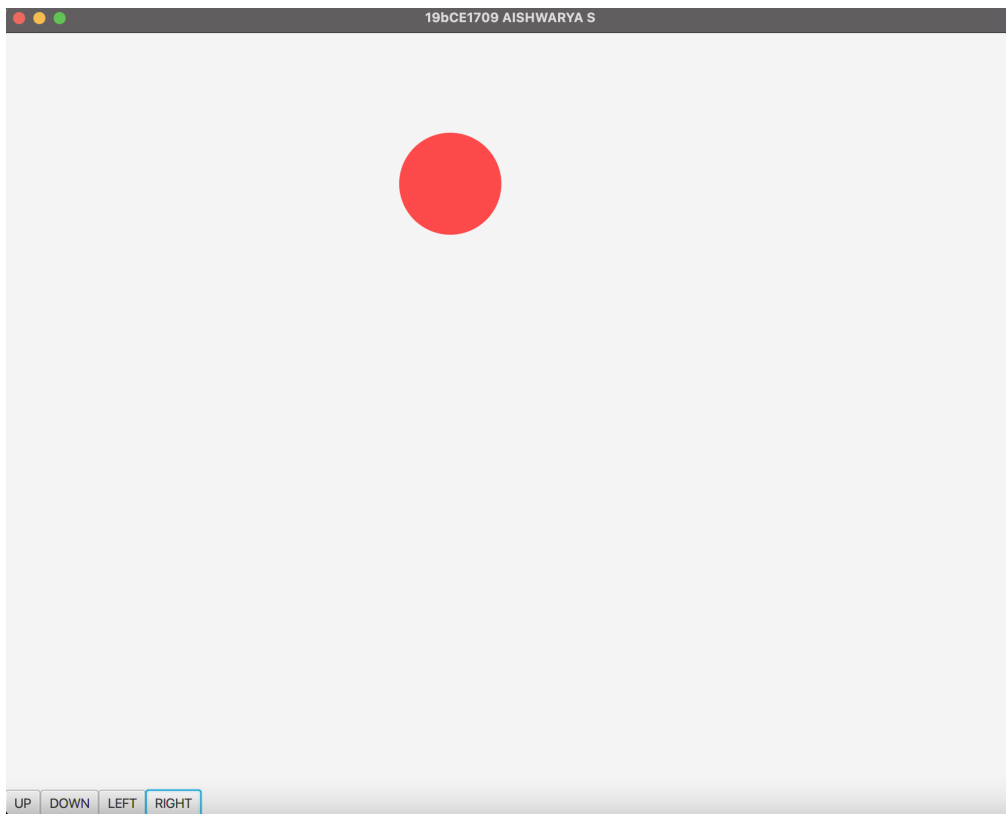
MOVING DOWN



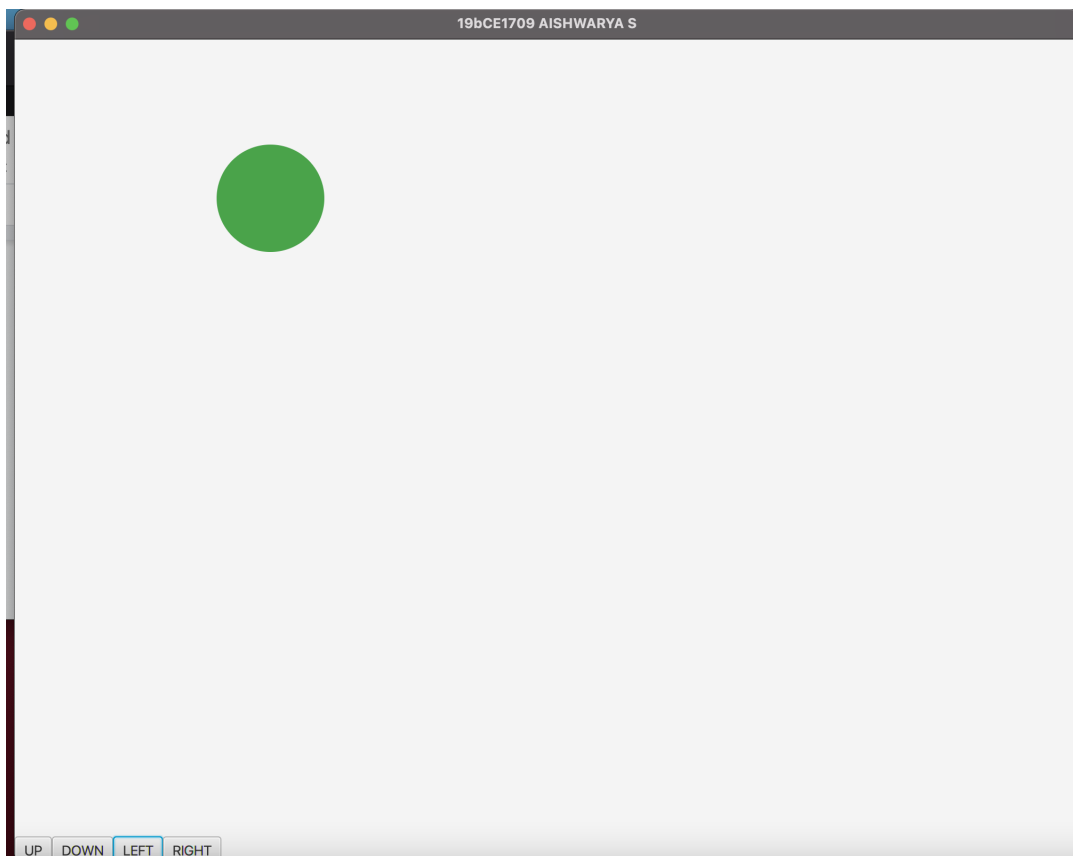
MOVING UP



MOVING RIGHT:



MOVING LEFT:



CODE:

```
package jlabs;

import javafx.application.Application;
import javafx.geometry.Bounds;

import javafx.scene.Scene;
import javafx.scene.control.Button;

import javafx.scene.layout.BorderPane;

import javafx.scene.layout.HBox;
import javafx.scene.layout.Pane;

import javafx.stage.Stage;
import javafx.scene.paint.Color;
import javafx.scene.shape.Circle;
public class lab11_2 extends Application {
    String[] sa= {"UP", "DOWN", "LEFT", "RIGHT"};
    Button[] b=new Button[4];
    static double x,y = 0;
    public static void main(String[] args) {
        launch(args);
    }
    public void start(Stage primaryStage) throws Exception {

        final Circle circle = new Circle(200, 150, 50, Color.BLUEVIOLET);
        circle.setTranslateX(0);
        circle.setTranslateY(0);

        circle.setOpacity(0.7);
        HBox hb= new HBox();
        for(int i=0;i<4;i++)
        {
            b[i]= new Button(sa[i]);
            hb.getChildren().add(b[i]);
        }
        BorderPane group = new BorderPane();

        Pane p= new Pane(circle);
        p.setPrefHeight(6000);
        p.setPrefWidth(6000);

        group.setBottom(hb);
        group.setCenter(p);
        Bounds bounds = p.getBoundsInLocal();

        b[1].setOnAction(e -> {
            y= circle.getLayoutY() + 40 ;

            if(y<=bounds.getMaxY()&&y>=bounds.getMinY())
            { y= circle.getLayoutY() + 40 ;
              circle.setLayoutY(y);
            }
        });
    }
}
```

```

        circle.setStroke(Color.BLACK);
        circle.setFill(Color.BLACK);

    }

});
b[0].setOnAction(e -> {

    y= circle.getLayoutY() - 40 ;

    if(y<=bounds.getMaxY()&&y>=bounds.getMinY())

    {

circle.setLayoutY(y);
circle.setStroke(Color.BLUE);
        circle.setFill(Color.DARKBLUE);
    }

});
b[2].setOnAction(e -> {
    x= circle.getLayoutX() - 40 ;
    if(x<=bounds.getMaxX()&&x>=bounds.getMinX())

    {

        circle.setLayoutX(x);
        circle.setStroke(Color.GREEN);
        circle.setFill(Color.GREEN);
    }

});
b[3].setOnAction(e -> {
    x= circle.getLayoutX() + 40 ;
    if(x<=bounds.getMaxX()&&x>=bounds.getMinX())

    {

        circle.setLayoutX(x);
        circle.setStroke(Color.RED);
        circle.setFill(Color.RED);
    }

});

```

```
});
```

```
Scene scene = new Scene(group,1000,1000,Color.LAVENDERBLUSH);  
primaryStage.setTitle("19bCE1709 AISHWARYA S");  
primaryStage.setScene(scene);  
primaryStage.show();
```

```
}  
}
```