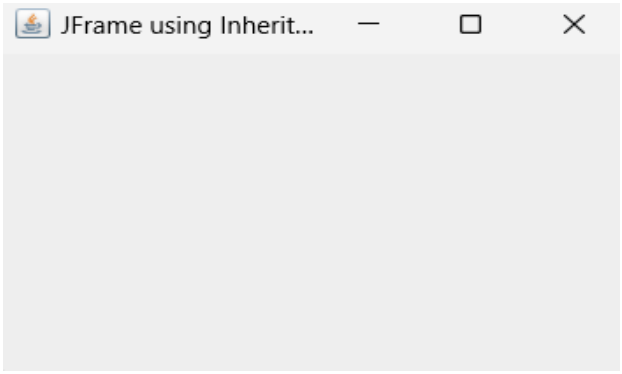
***Write a Java program to create an empty JFrame using Object and Inheritance concepts***

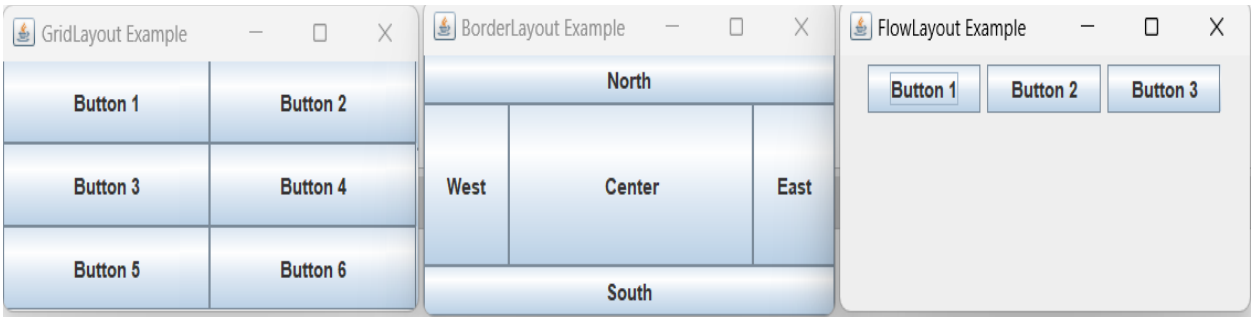
```
package emptyjframeusinginheritance;
import javax.swing.*;
class demoClass extends JFrame{
        demoClass(){
        setTitle("JFrame using Inheritance");
        setSize(300,200);
        setVisible(true);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        }
}
public class EmptyJFrameUsingInheritance {
    public static void main(String[] args) {
        new demoClass();
    }
}
```



***Demonstrate different layout managers.***

```
package layout.demonstration;
import javax.swing.*;
import java.awt.*;
class FlowLayoutDemo {
   public FlowLayoutDemo() {
      JFrame frame = new JFrame("FlowLayout Example");
      frame.setSize(400, 200);
      JPanel panel = new JPanel();
      panel.setLayout(new FlowLayout());
      panel.add(new JButton("Button 1"));
      panel.add(new JButton("Button 2"));
      panel.add(new JButton("Button 3"));
     frame.add(panel);
      frame.setVisible(true);
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

   }
}
class BorderLayoutDemo {
   public BorderLayoutDemo() {
      JFrame frame = new JFrame("BorderLayout Example");
      frame.setSize(400, 300);
      frame.setLayout(new BorderLayout());
      frame.add(new JButton("North"), BorderLayout.NORTH);
      frame.add(new JButton("South"), BorderLayout.SOUTH);
      frame.add(new JButton("East"), BorderLayout.EAST);
      frame.add(new JButton("West"), BorderLayout.WEST);
      frame.add(new JButton("Center"), BorderLayout.CENTER
      frame.setVisible(true);
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

   }
}
class GridLayoutDemo {
   public GridLayoutDemo() {
      JFrame frame = new JFrame("GridLayout Example");
      frame.setSize(400, 300);

      JPanel panel = new JPanel();
      panel.setLayout(new GridLayout(3, 2)); // 3 rows, 2 columns
      panel.add(new JButton("Button 1"));
      panel.add(new JButton("Button 2"));
      panel.add(new JButton("Button 3"));
      panel.add(new JButton("Button 4"));
      panel.add(new JButton("Button 5"));
      panel.add(new JButton("Button 6"));
      frame.add(panel);
      frame.setVisible(true);
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

   }
}
public class LayoutDemonstration {
   public static void main(String[] args) {
```

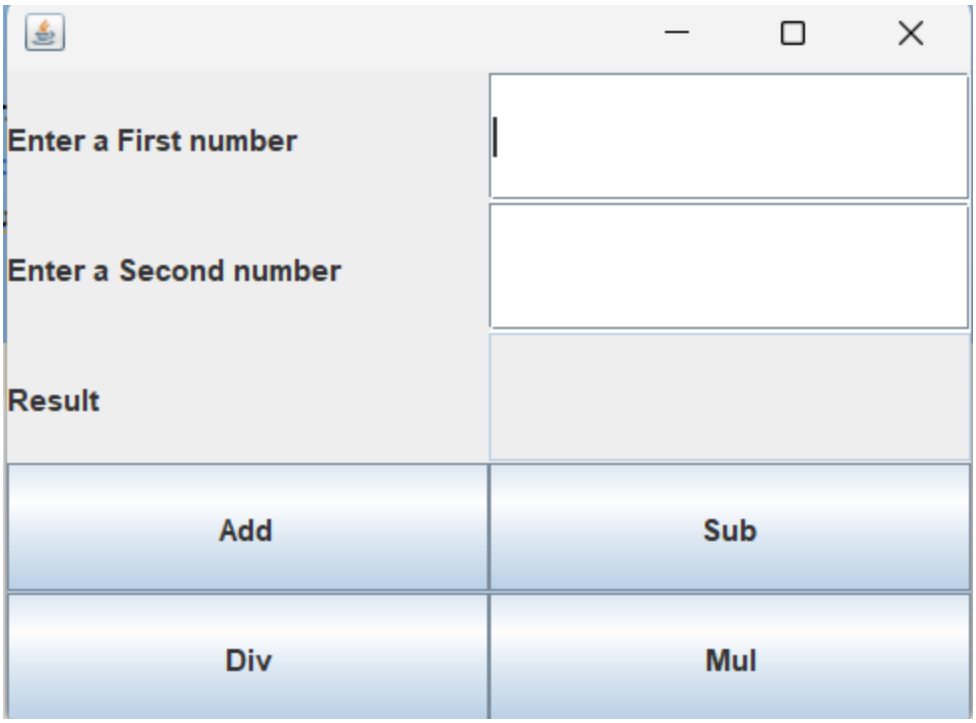
```
        new FlowLayoutDemo();
        new BorderLayoutDemo();
        new GridLayoutDemo();
    }
}
```



***c. Write a program using Swing components to perform arithmetic operation on two numbers. Use textfields for inputs and output. Your program should display the result when the user presses button Add, Subtract, Multiply and Divide***

```
package arthmetic.calculator;
import javax.swing.*;
import java.awt.*;

class Calculate {
   JFrame frame;
   JLabel num1, num2, result;
   JTextField num1TF, num2TF, resultTF;
   JButton add, mul, sub, div;
   int num11, num22; // Declare as instance variables

   Calculate() {
      frame = new JFrame();
      frame.setSize(400, 300);
      frame.setLayout(new GridLayout(5, 2));

      num1 = new JLabel("Enter a First number");
      frame.add(num1);
      num1TF = new JTextField(15);
      frame.add(num1TF);

      num2 = new JLabel("Enter a Second number");
      frame.add(num2);
      num2TF = new JTextField(15);
      frame.add(num2TF);

      result = new JLabel("Result");
      frame.add(result);
      resultTF = new JTextField(15);
      resultTF.setEditable(false); //Disable textField
      frame.add(resultTF);

      add = new JButton("Add");
      frame.add(add);

      sub = new JButton("Sub");
      frame.add(sub);

      div = new JButton("Div");
      frame.add(div);

      mul = new JButton("Mul");
      frame.add(mul);

      add.addActionListener(e -> ADD());
      sub.addActionListener(e -> SUB());
      div.addActionListener(e -> DIV());
      mul.addActionListener(e -> MUL());

      frame.setVisible(true);
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
   }
```

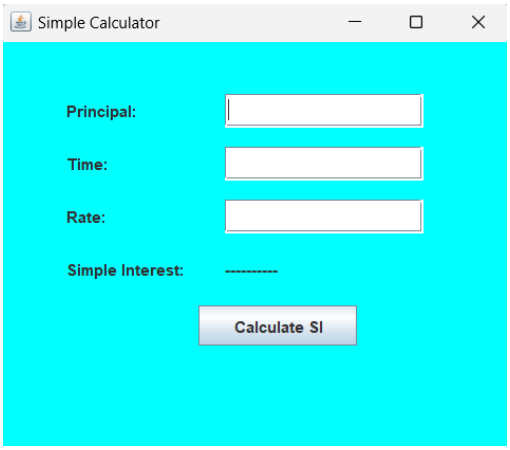
```
    public void getData() {
                num11 = Integer.parseInt(num1TF.getText());
            num22 = Integer.parseInt(num2TF.getText());
    }

    public void ADD() {
        getData ();
        int result = num11 + num22;
        resultTF.setText(String.valueOf(result));
    }

    public void SUB() {
        getData ();
        int result = num11 - num22;
        resultTF.setText(String.valueOf(result));
    }

    public void DIV() {
        getData ();
        if (num22 != 0) {
            int result = num11 / num22;
            resultTF.setText(String.valueOf(result));
        } else {
            JOptionPane.showMessageDialog(frame, "Cannot divide by zero.");
        }
    }

    public void MUL() {
        getData ();
        int result = num11 * num22;
        resultTF.setText(String.valueOf(result));
    }
}

public class ArthmeticCalculator {
    public static void main(String[] args) {
        new Calculate();
    }
}
```


Enter a First number
Enter a Second number
Result
Add
Sub
Div
Mul

***Write a program using a swing component to find simple interest. Use text fields for inputs and label for output. Your program should display the result when the user presses a button.***

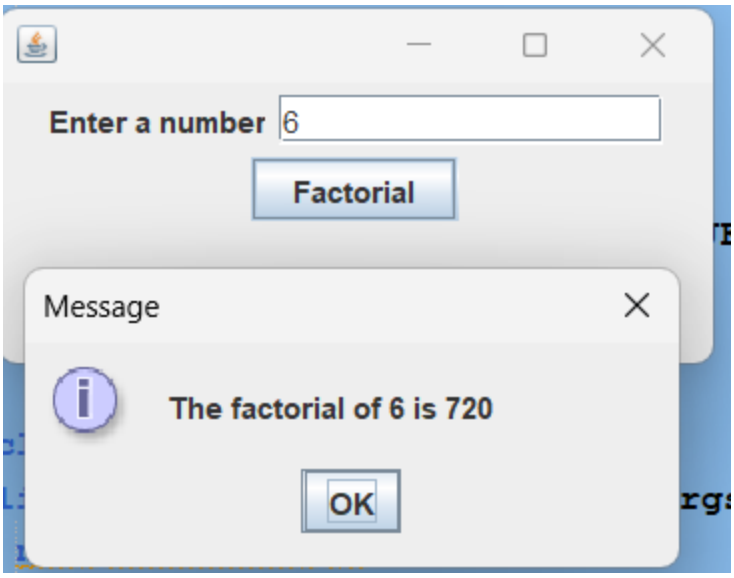
```
package simpleinterestusingnulllayout;

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

class InterestCaclulator
{
   /* Declaration of Object */
   JFrame frame;   /* Declaring object of JFrame.. So, that we can create a frame */
   JLabel lblPrincipal, lblTime, lblRate, lblResult, lblSi;
   JTextField txtPrincipal, txtTime, txtRate;
   JButton btnSi;
   JPanel panel;

   public InterestCaclulator()
   {
      /* Creating Object of JFrame */
      frame = new JFrame();
      frame.setTitle("Simple Calculator");
      frame.setSize(400, 350);
      frame.setLocationRelativeTo(null); /* Take frame to the Center Screen */
     // frame.setLayout(new FlowLayout());

      /* Creating and Adding Components */
      lblPrincipal = new JLabel("Principal: ");
      txtPrincipal = new JTextField(15);
      lblTime = new JLabel("Time: ");
      txtTime = new JTextField(15);
      lblRate= new JLabel("Rate: ");
      txtRate = new JTextField(15);
      lblResult = new JLabel("Simple Interest: ");
      lblSi = new JLabel("----------");
      btnSi = new JButton("Calculate SI");

     panel = new JPanel();
     panel.setBackground(Color.cyan);
     panel.setLayout(null);

     /* Setting the Boundary of Compoents on JPanel */
     lblPrincipal.setBounds(50, 40, 100, 25);
     txtPrincipal.setBounds(170, 40, 150,25);
     lblTime.setBounds(50, 80, 100, 25);
     txtTime.setBounds(170,80, 150, 25);
     lblRate.setBounds(50, 120, 100, 25);
     txtRate.setBounds(170, 120, 150, 25);
     lblResult.setBounds(50,160,100,25);
     lblSi.setBounds(170, 160, 100, 25);
     btnSi.setBounds(150,200,120,30);


      panel.add(lblPrincipal); panel.add(txtPrincipal);
      panel.add(lblTime); panel.add(txtTime);
```

```
        panel.add(lblRate); panel.add(txtRate);
        panel.add(lblResult); panel.add(lblSi);
        panel.add(btnSi);
        frame.add(panel);

        /*Anonymous or Inner Class for ActionEvent to calculate Simple Interest */
        btnSi.addActionListener(new ActionListener(){
           @Override
           public void actionPerformed(ActionEvent ae)
           {
              float p = Float.parseFloat(txtPrincipal.getText());
              float t = Float.parseFloat(txtTime.getText());
              float r = Float.parseFloat(txtRate.getText());

              float si = (p*t*r)/100;
              lblSi.setText(" "+si);

              JOptionPane.showMessageDialog(null, "Simple Interest = "+si);
           }
        });

        /* Lambda Expression for Anonymous Event */
//        btnSi.addActionListener((ActionEvent ae) -> {
//           //Code goes here
//        });

            /* Make frame visible */
        frame.setVisible(true);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

/* Driver Class */
public class SimpleInterestUsingNullLayout {

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



***Write a GUI program to compute factorial of number. Use textfield for input and pre-defined dialog box for output and one button which upon mouse click computes factorial of a number.***
***package factorialnumbers;***

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

class Factorial {
   JFrame frame;
   JLabel num;
   JTextField numTF;
   JButton factorialButton;

   Factorial() {
      frame = new JFrame();
      frame.setSize(300, 150);
      frame.setLayout(new FlowLayout());

      num = new JLabel("Enter a number");
      frame.add(num);

      numTF = new JTextField(15);
      frame.add(numTF);

      factorialButton = new JButton("Factorial");
      frame.add(factorialButton);

      factorialButton.addActionListener((ActionEvent e) -> {

            int number = Integer.parseInt(numTF.getText());
            if (number < 0) {
               JOptionPane.showMessageDialog(frame, "Factorial is not defined for negative numbers.");
            } else {
               long res = 1;
               for (int i = 1; i <= number; ++i) {
                  res *= i;
               }
               JOptionPane.showMessageDialog(frame, "The factorial of " + number + " is " + res);
            }

      });

      frame.setVisible(true);
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
   }
}

public class FactorialNumbers {
   public static void main(String[] args) {
      new Factorial();
   }
}
```


Enter a number
6
Factorial
Message
The factorial of 6 is 720
OK

***f. Demonstrate MouseEvent and KeyEvent using both Event Interface and Adapter class.***

***Mouse Event  with Event Interface***

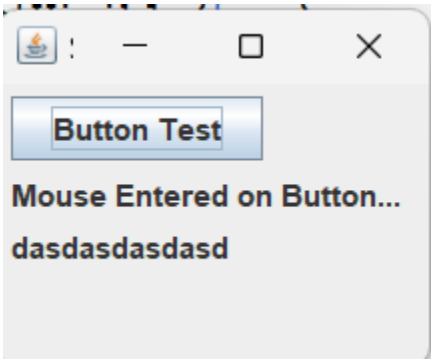
```
package mouseevent.withevent Interface;

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

class event implements MouseListener{
   JFrame frame;
   JButton btnTest;
   JLabel lblOutput;
   event(){
      frame = new JFrame();
      frame.setTitle("Simple Calculator");
      frame.setSize(200, 100);
      frame.setLocationRelativeTo(null); /* Take frame to the Center Screen */
      frame.setLayout(new FlowLayout(FlowLayout.LEFT));

      //Creating and Adding GUI Controls
      btnTest = new JButton("Button Test");
      frame.add(btnTest);
      lblOutput = new JLabel("--------------------");
      frame.add(lblOutput);

      /* Add Mouse Event to Button */
      btnTest.addMouseListener(this);


       /* Make frame visible */
      frame.setVisible(true);
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

   }

   /* Mouse Event methods outside Construcutor */
   @Override
   public void mouseClicked(MouseEvent e) {
     lblOutput.setText("Mouse Clicked on Button....");
   }

   @Override
   public void mousePressed(MouseEvent e) {
      lblOutput.setText("Mouse Pressed on Button...");
   }

   @Override
   public void mouseReleased(MouseEvent e) {
       lblOutput.setText("Mouse Released from Button...");
   }

   @Override
   public void mouseEntered(MouseEvent e) {
       lblOutput.setText("Mouse Entered on Button...");
```

```
    }

    @Override
    public void mouseExited(MouseEvent e) {
        lblOutput.setText("Mouse Exited from Button...");
    }
}


class MouseEventWithEventInterface {
    public static void main(String[] args) {
        new event();
    }
}
```

**Mouse Event  with Adapter Class**

```
package mouseeventbwic_1;

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

class MouseEvent extends MouseAdapter{
    JFrame frame;
    JButton btnTest;
    JLabel lblOutput,res ;

    public MouseEvent () {
        // Creating Object of JFrame
        frame = new JFrame();
        frame.setTitle("Simple Calculator");
        frame.setSize(150, 150);
        frame.setLocationRelativeTo(null); // Center the frame
        frame.setLayout(new FlowLayout(FlowLayout.LEFT));

        // Creating and Adding GUI Controls
        btnTest = new JButton("Button Test");
        frame.add(btnTest);
        lblOutput = new JLabel("--------------------");
        frame.add(lblOutput);
        res = new JLabel("dasdasdasdasd ");
        frame.add(res);

        // Add Mouse Event to Button using MouseAdapter for simplicity
        btnTest.addMouseListener(new MouseAdapter() {
          @Override
          public void mouseEntered(MouseEvent e) {
              lblOutput.setText("Mouse Entered on Button...");
          }
          @Override
          public void mousePressed(MouseEvent e) {
              res.setText("Mouse Pressed on Button...");
          }
          @Override
          public void mouseExited(MouseEvent e) {
```

```
            Toolkit.getDefaultToolkit().beep();
        }
      });

      // Make frame visible
      frame.setVisible(true);
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
   }
}

/* Driver Class */
public class MouseEventBWIC_1 {
   public static void main(String[] args) {
      new MouseEvent();
   }
}
```




**Key Event with Key Interface**

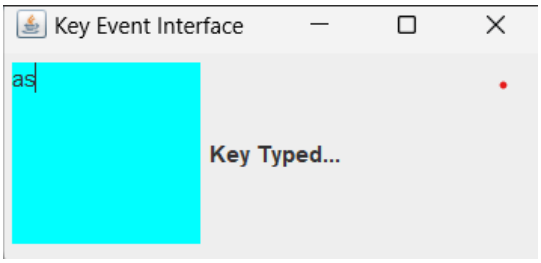
```
package keyevent_keylistener;

import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import javax.swing.*;
class KeyEventInterface implements KeyListener
{
   JFrame frame;
   JTextArea jta;
   JLabel lblOutput;

   public KeyEventInterface()
   {
       /* Creating Object of JFrame */
      frame = new JFrame();
      frame.setTitle("Key Event Interface");
      frame.setSize(300, 150);
      frame.setLocationRelativeTo(null); /* Take frame to the Center Screen */
      frame.setLayout(new FlowLayout(FlowLayout.LEFT));

      jta = new JTextArea(6,10);
      jta.setBackground(Color.cyan);
```

```
        frame.add(jta);
        lblOutput = new JLabel("--------------------");
        frame.add(lblOutput);

        /*Add key event to textarea i.e. jta*/
        jta.addKeyListener(this);
 /* Make frame visible */
        frame.setVisible(true);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

    /* Override key event methods outside Constructor */
    @Override
    public void keyPressed(KeyEvent k)
    {
        lblOutput.setText("Key has been Pressed...");
    }

    public void keyTyped(KeyEvent k)
    {

       lblOutput.setText("Key Typed...");
    }

    public void keyReleased(KeyEvent ke)
    {
        lblOutput.setText("Key Released...");
    }
}

public class KeyEvent_keyListener {

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

}
```

**Key Event with Key Adapter**

```
package keyevent_keyadapter;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
class KeyAdapterClass extends KeyAdapter
{
    JFrame frame;
    JTextArea jta;
    JLabel lblOutput;

    public KeyAdapterClass()
    {
         /* Creating Object of JFrame */
        frame = new JFrame();
        frame.setTitle("Key Event Adapter Class ");
```

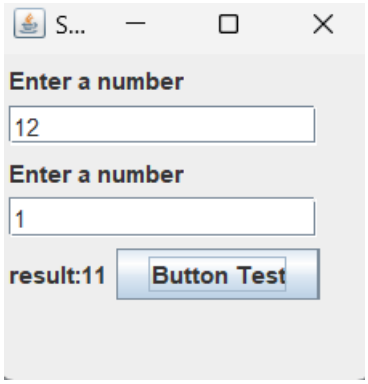
```
        frame.setSize(300, 150);
        frame.setLocationRelativeTo(null); /* Take frame to the Center Screen */
        frame.setLayout(new FlowLayout(FlowLayout.LEFT));

        jta = new JTextArea(6,10);
        jta.setBackground(Color.cyan);
        frame.add(jta);
        lblOutput = new JLabel("--------------------");
        frame.add(lblOutput);
              /*Add key event to textarea i.e. jta*/
        jta.addKeyListener(this);
        /* Make frame visible */
        frame.setVisible(true);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
      /* Override key event methods outside Constructor */
    @Override
    public void keyPressed(KeyEvent k)
    {
      lblOutput.setText("Key has been Pressed...");
    }

    public void keyReleased(KeyEvent ke)
    {
      lblOutput.setText("Key Released...");
    }
}
public class KeyEvent_keyAdapter {

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



***g. Write a GUI program using components to find sum and difference of two numbers. Use two textfields for giving input and a label for output. The program should display sum if user presses mouse and difference if user release mouse.***

```
package lab1_q_no_g;
import javax.swing.*;
import java.awt.*;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;

class event extends MouseAdapter{
   JFrame frame;
   JButton btnTest;
   JLabel txt1,txt2 , result ;
   JTextField num1,num2;

   public event() {
      // Creating Object of JFrame
      frame = new JFrame();
      frame.setTitle("Simple Calculator");
      frame.setSize(200, 200);
      frame.setLocationRelativeTo(null); // Center the frame
      frame.setLayout(new FlowLayout(FlowLayout.LEFT));

      txt1 = new JLabel("Enter a number");
      frame.add(txt1);

      num1 = new JTextField(15);
      frame.add(num1);

      txt2 = new JLabel("Enter a number");
      frame.add(txt2);

      num2 = new JTextField(15);
      frame.add(num2);

        result = new JLabel("result: ");
      frame.add(result);

      // Creating and Adding GUI Controls
      btnTest = new JButton("Button Test");
      frame.add(btnTest);

      // Add Mouse Event to Button using MouseAdapter for simplicity
      btnTest.addMouseListener(new MouseAdapter() {

         @Override
         public void mousePressed(MouseEvent e) {
            int a = Integer.parseInt(num1.getText());
            int b = Integer.parseInt(num2.getText());
            int c = a+b;
           result.setText(String.valueOf("result:"+c));

         }
         @Override
```

```
            public void mouseExited(MouseEvent e) {
               int a = Integer.parseInt(num1.getText());
                int b = Integer.parseInt(num2.getText());
                int c = a-b;
               result.setText(String.valueOf("result:"+c));
            }
        });

        // Make frame visible
        frame.setVisible(true);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

public class Lab1_Q_no_g {

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

}
```

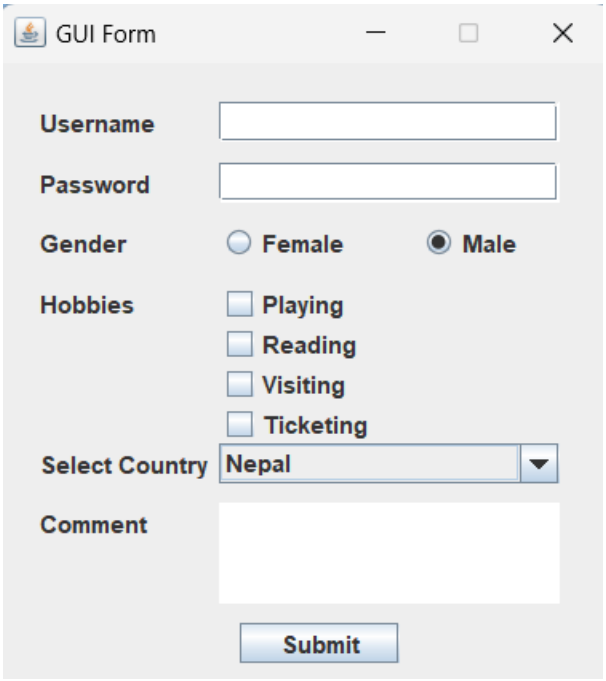


***h. Demonstrate different GUI controls available in Swing using Null layout manager.***

```
package guicontrolsform;
import javax.swing.*;
import java.awt.*;

class guiclass{
   JFrame frame;
   JLabel lblUser, lblPass, lblGender, lblHobbies, lblCountry, lblComment;
   JTextField txtUser;
   JPasswordField jpfPass;

   JRadioButton rdMale, rdFemale;
   ButtonGroup rdGroup;  // to grooup male and female

   JCheckBox chk1, chk2, chk3, chk4;

   JComboBox jcb;
   JTextArea jta;
   JButton btnSubmit;

   guiclass(){
      frame = new JFrame();
      frame.setTitle("GUI Form");
      frame.setSize(320,350);
      frame.setLayout(null);
      frame.setResizable(false);
      frame.setLocation(580,180);


      lblUser = new JLabel("Username");
      lblUser.setBounds(20,20,110,20);
      frame.add(lblUser);

      txtUser = new JTextField();
      txtUser.setBounds(110,20,170,20);
      frame.add(txtUser);

       lblPass = new JLabel("Password");
      lblPass.setBounds(20,50,110,20);
      frame.add(lblPass);

      jpfPass = new JPasswordField();
      jpfPass.setBounds(110,50,170,20);
      frame.add(jpfPass);

        lblGender = new JLabel("Gender");
      lblGender.setBounds(20,80,110,20);
      frame.add(lblGender);

//       radio button
      rdFemale = new JRadioButton("Female");
      rdFemale.setBounds(110,80,100,20);
      frame.add(rdFemale);

      rdMale = new JRadioButton("Male",true);
      rdMale.setBounds(210,80,110,20);
```

```
        frame.add(rdMale);

        rdGroup = new ButtonGroup();
        rdGroup.add(rdMale);
        rdGroup.add(rdFemale);

         lblHobbies = new JLabel("Hobbies");
        lblHobbies.setBounds(20,110,110,20);
        frame.add(lblHobbies);

//       Check box
        chk1 = new JCheckBox("Playing");
        chk1.setBounds(110,110,170,20);
        frame.add(chk1);

        chk2 = new JCheckBox("Reading");
        chk2.setBounds(110,130,170,20);
        frame.add(chk2);
        chk3 = new JCheckBox("Visiting");
        chk3.setBounds(110,150,170,20);
        frame.add(chk3);

        chk4 = new JCheckBox("Ticketing");
        chk4.setBounds(110,170,170,20);
        frame.add(chk4);


         lblCountry = new JLabel("Select Country");
        lblCountry.setBounds(20,190,110,20);
        frame.add(lblCountry);

//       JComboBox
        String[] countryList = {"Nepal","India","China","Other"};
        jcb = new JComboBox(countryList);
        jcb.setBounds(110,190,170,20);
        frame.add(jcb);

        lblComment = new JLabel("Comment");
        lblComment.setBounds(20,220,110,20);
        frame.add(lblComment);

//       JTextArea
        jta = new JTextArea();
        jta.setBounds(110,220,170,50);
        frame.add(jta);

        btnSubmit = new JButton("Submit");
        btnSubmit.setBounds(120,280,80,20);

        frame.add(btnSubmit);

        frame.setVisible(true);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```

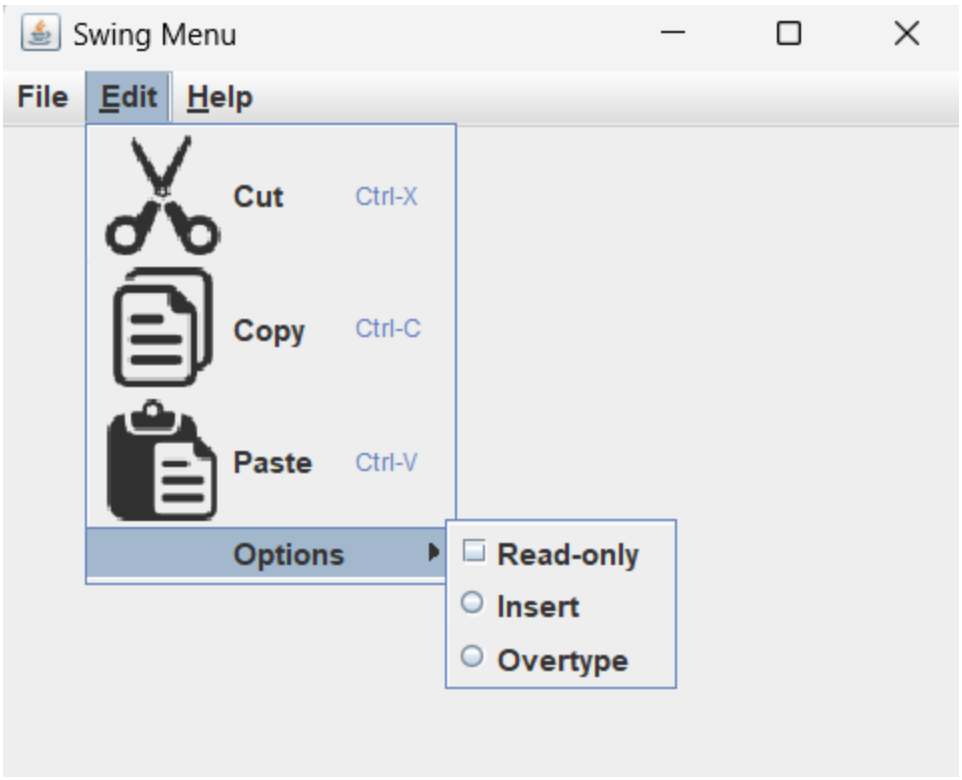
```
public class GuiControlsForm {
   public static void main(String[] args) {
      new guiclass();
   }

}
```



***i.. Create a MenuBar with Some Menus and Menuitems using Swing and Even***

```
package menubar;
//import java.awt.Image;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;
import javax.swing.*;
class MenuBardemo{
    JFrame frame;
   JMenuBar menubar;
   JMenu fileMenu, editMenu, helpMenu,optionsMenu;
   JMenuItem cutItem, copyItem, pasteItem;
   JCheckBoxMenuItem chkReadonlyItem;
   JRadioButtonMenuItem rdInsertItem, rdOvertypeItem;
   ButtonGroup rdGroup;
   MenuBardemo(){
      frame = new JFrame();
      frame.setTitle("Swing Menu");
      frame.setSize(400, 200);
      frame.setLocationRelativeTo(null); /* Take frame to the Center Screen */
      frame.setLayout(null);
      //creating  object of menu
     menubar = new JMenuBar();
     //Creating menus so that we can add it to memubar
     fileMenu = new JMenu("File");
     menubar.add(fileMenu);
      editMenu = new JMenu("Edit");
     menubar.add(editMenu);
      helpMenu = new JMenu("Help");
     menubar.add(helpMenu);


      /* Create Menu items and add to respective menu */
      cutItem = new JMenuItem("Cut", new ImageIcon(getClass().getResource("/iconpack/cut.png")));
      editMenu.add(cutItem);
     copyItem = new JMenuItem("Copy", new ImageIcon(getClass().getResource("/iconpack/copy.png")));
      editMenu.add(copyItem);
      pasteItem = new JMenuItem("Paste", new ImageIcon(getClass().getResource("/iconpack/paste.png")));
      editMenu.add(pasteItem);
      optionsMenu = new JMenu("Options");
       editMenu.add(optionsMenu);

        /* Creating checkbox menu item and add to optionsMenu */
      chkReadonlyItem = new JCheckBoxMenuItem("Read-only");
      optionsMenu.add(chkReadonlyItem);

      /* Creating radio button menu item and add to optionsMenu */
       rdInsertItem = new JRadioButtonMenuItem("Insert");
       optionsMenu.add(rdInsertItem);
      rdOvertypeItem = new JRadioButtonMenuItem("Overtype");
       optionsMenu.add(rdOvertypeItem);
        rdGroup = new ButtonGroup();
      rdGroup.add(rdInsertItem);
      rdGroup.add(rdOvertypeItem);
```

```
        /* Setting Mnenomics */
        helpMenu.setMnemonic('H');
        editMenu.setMnemonic('E');

        /* Setting accelator to respective menu item */
        cutItem.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_X, ActionEvent.CTRL_MASK));
         copyItem.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_C, ActionEvent.CTRL_MASK));

        pasteItem.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_V, ActionEvent.CTRL_MASK));


        JMenuItem exitItem = new JMenuItem("Exit");
        fileMenu.add(exitItem);

        /*Anonymous ActionEvent on exitItem without Lambda */
        exitItem.addActionListener(new ActionListener()
        {
           public void actionPerformed(ActionEvent ae)
           {
              System.exit(0);
           }
        });

        /* Anonymouse Event using Lambda expression */
        cutItem.addActionListener((e)->
        {
           String str = e.getActionCommand();
           JOptionPane.showMessageDialog(frame, "You have selected -> "+str);
        });

        /* Adding menubar to JFrame using setJMenuBar() */
        frame.setJMenuBar(menubar);

        frame.setVisible(true);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    }
}
public class MenuBar {

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

}
```


Swing Menu
File
Edit
Help
Cut
Ctrl-X
Copy
Ctrl-C
Paste
Ctrl-V
Options
Read-only
Insert
Overtype

```
/* Step 1: Import the required packages */
import java.sql.*;

public class ConsoleCrudBWIC {
   public static void main(String[] args) throws SQLException, ClassNotFoundException {
      /*Step 2: Load and Register the MySQL Driver */
      Class.forName("com.mysql.jdbc.Driver");
      /*Step 3: Create and Establish the Connection with MySQL Server*/
      String URL = "jdbc:mysql://localhost:3306/bridgewaterdb";
      String username="root";
      String password="";
      Connection conn = DriverManager.getConnection(URL, username, password);

      /*Step 4: Create a Statement */
      Statement stmt = conn.createStatement();

      /*Step 5: Create and Execute SQL statment using above stmt */
      String createTable = "CREATE Table Student (id int auto_increment primary key,"
            + "name varchar(50),"
            + "age int)";
      stmt.execute(createTable);

      /* Step 6: Check or Process the above query is executed or not */
      System.out.println("Table created successfully....");

      /* Step 7: Close the connection */
      conn.close();
   }
}
```

***j. Provided that a MySQL database named "BookDb" with table named "Book_Info" with columns (id as int, title as varchar(20), author as varchar(20), publication as varchar(20) and price as int). Write a java program to connect to the database named BookDb with the table Book_Info and perform the console CRUD operation.***

```
package lab1_j_code_crud;

import java.sql.*;
import java.util.Scanner;

public class Lab1_j_code_crud {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        // Displaying the CRUD menu
        System.out.println("--------Console CRUD Menu---------");
        System.out.println("1. Insert");
        System.out.println("2. Update");
        System.out.println("3. Delete");
        System.out.println("4. Select");
        System.out.println("5. Exit");
        System.out.println("---------------------------------");

        int choice;

        do {
            System.out.println("Enter your choice from menu:");
            choice = input.nextInt();

            switch (choice) {
                case 1:
                    insertRecord();
                    break;
                case 2:
                    updateRecord();
                    break;
                case 3:
                    deleteRecord();
                    break;
                case 4:
                    selectRecord();
                    break;
                case 5:
                    System.out.println("Exiting program...");
                    break;
                default:
                    System.out.println("Invalid choice!!! Try again.....");
            }
        } while (choice != 5);

        input.close();
    }

    private static void insertRecord() {
        try {
```

```
            // Establishing database connection
            Connection conn = getConnection();

            // Creating statement
            Statement stmt = conn.createStatement();

            // SQL query for insert
            String insert = "INSERT INTO Book_Info (id, title, author, publication, price) " +
                            "VALUES (1, 'Fire And Blood', 'sumaan', 'HBO', 500)";

            // Executing insert query
            stmt.executeUpdate(insert);

            // Closing resources
            stmt.close();
            conn.close();

            System.out.println("Record Inserted Successfully");
        } catch (SQLException e) {
            System.out.println("SQL Exception: " + e.getMessage());
        }
    }

    private static void updateRecord() {
        try {
            // Establishing database connection
            Connection conn = getConnection();

            // Creating statement
            Statement stmt = conn.createStatement();

            // SQL query for update
            String update = "UPDATE Book_Info SET title = 'fidasdasdasdre', " +
                            "author = 'khaire', publication = 'kantipur', price = 456 " +
                            "WHERE id = 41";

            // Executing update query
            stmt.executeUpdate(update);

            // Closing resources
            stmt.close();
            conn.close();

            System.out.println("Record Updated Successfully");
        } catch (SQLException e) {
            System.out.println("SQL Exception: " + e.getMessage());
        }
    }

    private static void deleteRecord() {
        try {
            // Establishing database connection
            Connection conn = getConnection();

            // Creating statement
            Statement stmt = conn.createStatement();
```

```
        // SQL query for delete
        String delete = "DELETE FROM Book_Info WHERE id = 1";

        // Executing delete query
        stmt.executeUpdate(delete);

        // Closing resources
        stmt.close();
        conn.close();

        System.out.println("Record Deleted Successfully");
    } catch (SQLException e) {
        System.out.println("SQL Exception: " + e.getMessage());
    }
}

private static void selectRecord() {
    try {
        // Establishing database connection
        Connection conn = getConnection();

        // Creating statement
        Statement stmt = conn.createStatement();

        // SQL query for select
        String selectQuery = "SELECT * FROM Book_Info";

        // Executing select query
        ResultSet rs = stmt.executeQuery(selectQuery);

        // Printing results
        System.out.println("ID\tTitle\t\tAuthor\t\tPublication\tPrice");
        while (rs.next()) {
            int id = rs.getInt("id");
            String title = rs.getString("title");
            String author = rs.getString("author");
            String pub = rs.getString("publication");
            int price = rs.getInt("price");
            System.out.println(id + "\t" + title + "\t\t" + author + "\t\t" + pub + "\t\t" + price);
        }

        // Closing resources
        rs.close();
        stmt.close();
        conn.close();

        System.out.println("Record Selected Successfully");
    } catch (SQLException e) {
        System.out.println("SQL Exception: " + e.getMessage());
    }
}

private static Connection getConnection() throws SQLException {
    String url = "jdbc:mysql://localhost:3306/BookDb";
    String username = "root";
```

```
        String password = "";

        // Registering JDBC driver
        try {
            Class.forName("com.mysql.jdbc.Driver");
        } catch (ClassNotFoundException e) {
            System.out.println("ClassNotFoundException: " + e.getMessage());
        }

        // Returning connection object
        return DriverManager.getConnection(url, username, password);
    }
}
```

---

```
package lab1_j_code_crud;

import java.sql.*;
import java.util.Scanner;

public class Lab1_j_code_crud {

    public static void main(String[] args) {


        // Displaying the CRUD menu
        System.out.println("--------Console CRUD Menu---------");
        System.out.println("1. Insert");
        System.out.println("2. Update");
        System.out.println("3. Delete");
        System.out.println("4. Select");
        System.out.println("5. Exit");
        System.out.println("---------------------------------");

        int choice;

      Scanner input = new Scanner(System.in);
            System.out.println("Enter your choice from menu:");
            choice = input.nextInt();

            switch (choice) {
                case 1:
                    insertRecord();
                    break;
                case 2:
                    updateRecord();
                    break;
                case 3:
                    deleteRecord();
                    break;
                case 4:
                    selectRecord();
                    break;
                case 5:
```

```
            System.out.println("Exiting program...");
            break;
        default:
            System.out.println("Invalid choice!!! Try again.....");
    }


}

private static void insertRecord() {
    try {
        // Establishing database connection
        Connection conn = getConnection();

        // Creating statement
        Statement stmt = conn.createStatement();

        // SQL query for insert
        String insert = "INSERT INTO Book_Info (id, title, author, publication, price) " +
                        "VALUES (1, 'Fire And Blood', 'sumaan', 'HBO', 500)";

        // Executing insert query
        stmt.executeUpdate(insert);

        // Closing resources
        stmt.close();
        conn.close();

        System.out.println("Record Inserted Successfully");
    } catch (SQLException e) {
        System.out.println("SQL Exception: " + e.getMessage());
    }
}

private static void updateRecord() {
    try {
        // Establishing database connection
        Connection conn = getConnection();

        // Creating statement
        Statement stmt = conn.createStatement();

        // SQL query for update
        String update = "UPDATE Book_Info SET title = 'fidasdasdasdre', " +
                        "author = 'khaire', publication = 'kantipur', price = 456 " +
                        "WHERE id = 41";

        // Executing update query
        stmt.executeUpdate(update);

        // Closing resources
        stmt.close();
        conn.close();

        System.out.println("Record Updated Successfully");
    } catch (SQLException e) {
```

```
            System.out.println("SQL Exception: " + e.getMessage());
        }
    }

    private static void deleteRecord() {
        try {
            // Establishing database connection
            Connection conn = getConnection();

            // Creating statement
            Statement stmt = conn.createStatement();

            // SQL query for delete
            String delete = "DELETE FROM Book_Info WHERE id = 1";

            // Executing delete query
            stmt.executeUpdate(delete);

            // Closing resources
            stmt.close();
            conn.close();

            System.out.println("Record Deleted Successfully");
        } catch (SQLException e) {
            System.out.println("SQL Exception: " + e.getMessage());
        }
    }

    private static void selectRecord() {
        try {
            // Establishing database connection
            Connection conn = getConnection();

            // Creating statement
            Statement stmt = conn.createStatement();

            // SQL query for select
            String selectQuery = "SELECT * FROM Book_Info";

            // Executing select query
            ResultSet rs = stmt.executeQuery(selectQuery);

            // Printing results
            System.out.println("ID\tTitle\t\tAuthor\t\tPublication\tPrice");
            while (rs.next()) {
                int id = rs.getInt("id");
                String title = rs.getString("title");
                String author = rs.getString("author");
                String pub = rs.getString("publication");
                int price = rs.getInt("price");
                System.out.println(id + "\t" + title + "\t\t" + author + "\t\t" + pub + "\t\t" + price);
            }

            // Closing resources
            rs.close();
            stmt.close();
```

```
            conn.close();

            System.out.println("Record Selected Successfully");
        } catch (SQLException e) {
            System.out.println("SQL Exception: " + e.getMessage());
        }
    }

    private static Connection getConnection() throws SQLException {
        String url = "jdbc:mysql://localhost:3306/BookDb";
        String username = "root";
        String password = "";

        // Registering JDBC driver
        try {
            Class.forName("com.mysql.jdbc.Driver");
        } catch (ClassNotFoundException e) {
            System.out.println("ClassNotFoundException: " + e.getMessage());
        }

        // Returning connection object
        return DriverManager.getConnection(url, username, password);
    }
}
```

ut - **lab1_j_code_crud (run) #4** ×

```
run:
--------Console CRUD Menu---------
1. Insert
2. Update
3. Delete
4. Select
5. Exit
---------------------------------
Enter your choice from menu:
4
ID      Title           Author          Publication     Price
2       Fire            John            HBO             600
Record Selected Successfully
Enter your choice from menu:
1
Record Inserted Successfully
Enter your choice from menu:
4
ID      Title           Author          Publication     Price
1       Fire            suman           HBO             500
2       Fire            John            HBO             600
Record Selected Successfully
Enter your choice from menu:
2
Record Updated Successfully
Enter your choice from menu:
4
ID      Title           Author          Publication     Price
1       dragon          khaire          HBO             456
2       Fire            John            HBO             600
Record Selected Successfully
Enter your choice from menu:
3
Record Deleted Successfully
Enter your choice from menu:
4
ID      Title           Author          Publication     Price
2       Fire            John            HBO             600
Record Selected Successfully
```

***k. Create book entry form (id, title, author, publication, price) and Insert, Update, Delete and View records in/from database using JDBC and Swing or Perform GUI CRUD operations***

**## Advance**

```
package bookentryform;

import java.sql.*;
import javax.swing.*;
import javax.swing.table.*;

class BEF {

    JFrame frame;
    JLabel title, author, publication, price;
    JTextField title_txt, author_txt, publication_txt, price_txt;
    JButton update, insert, delete, view;

    JTable table;
    JScrollPane sp;
    DefaultTableModel tblModel;

    BEF() {
        frame = new JFrame();
        frame.setSize(405, 400);
        frame.setLayout(null);
        frame.setTitle("Book Management System");
        frame.setResizable(true);

        title = new JLabel("Title");
        title.setBounds(20, 20, 80, 20);
        frame.add(title);

        title_txt = new JTextField();
        title_txt.setBounds(105, 20, 265, 20);
        frame.add(title_txt);

        author = new JLabel("Author");
        author.setBounds(20, 40, 80, 20);
        frame.add(author);

        author_txt = new JTextField();
        author_txt.setBounds(105, 40, 265, 20);
        frame.add(author_txt);

        publication = new JLabel("Publication");
        publication.setBounds(20, 60, 80, 25);
```

```
        frame.add(publication);

        publication_txt = new JTextField();
        publication_txt.setBounds(105, 60, 265, 20);
        frame.add(publication_txt);

        price = new JLabel("Price");
        price.setBounds(20, 80, 80, 25);
        frame.add(price);

        price_txt = new JTextField();
        price_txt.setBounds(105, 80, 265, 20);
        frame.add(price_txt);

        insert = new JButton("Insert");
        insert.setBounds(20, 100, 80, 20);
        frame.add(insert);

        update = new JButton("Update");
        update.setBounds(110, 100, 80, 20);
        frame.add(update);

        delete = new JButton("Delete");
        delete.setBounds(200, 100, 80, 20);
        frame.add(delete);

        view = new JButton("View");
        view.setBounds(290, 100, 80, 20);
//        frame.add(view);

        /* Creating JTable to add row from database table */
        String[] columns = {"ID", "Title", "Author", "Publication", "Price"};

        tblModel = new DefaultTableModel(columns, 0);
        table = new JTable(tblModel);
        sp = new JScrollPane(table);
        frame.add(sp);
        sp.setBounds(20, 200, 350, 150);

        selectRecord();

        insert.addActionListener(e -> insertRecord());
        delete.addActionListener(e -> deleteRecord());
        update.addActionListener(e -> updateRecord());
        view.addActionListener(e -> selectRecord());
```

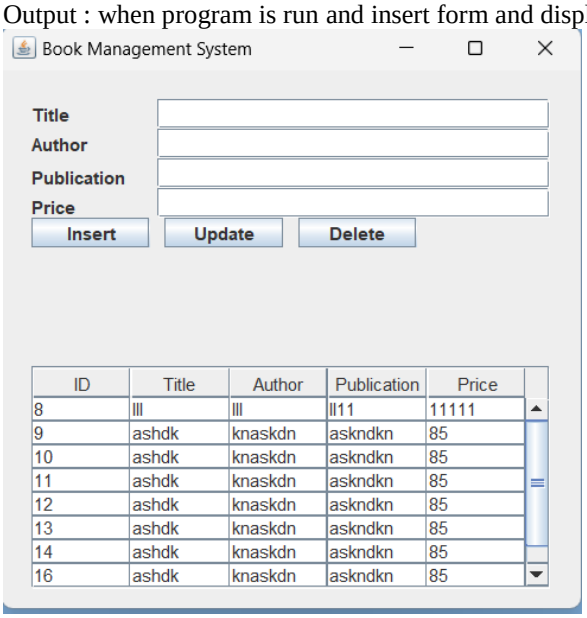
```
        frame.setVisible(true);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    }

    public void insertRecord() {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            String URL = "jdbc:mysql://localhost:3306/bookdb";
            String username = "root";
            String password = "";
            Connection conn = DriverManager.getConnection(URL, username,
password);
            Statement stmt = conn.createStatement();

            String titleD = title_txt.getText();
            String authorD = author_txt.getText();
            String pubD = publication_txt.getText();
            int priceD = Integer.parseInt(price_txt.getText());

            if (!titleD.isEmpty() && !authorD.isEmpty() && !pubD.isEmpty() &&
priceD >0 ) {
                String insertQuery = "INSERT INTO book_info (title, author,
publication, price) VALUES ('" + titleD + "', '" + authorD + "', '" + pubD + "',
'" + priceD + "')";
                stmt.execute(insertQuery);
                JOptionPane.showMessageDialog(frame, "Record inserted
successfully.");
                selectRecord();
            } else {
                JOptionPane.showMessageDialog(frame, "Empty TextField");
                selectRecord();
            }

        } catch (Exception e) {
            JOptionPane.showMessageDialog(frame, "Error: " + e.getMessage());
        }
    }

    public void deleteRecord() {
        String n = JOptionPane.showInputDialog("Enter ID number for delete:");

        try {
            Class.forName("com.mysql.jdbc.Driver");
```

```
            String URL = "jdbc:mysql://localhost:3306/bookdb";
            String username = "root";
            String password = "";
            Connection conn = DriverManager.getConnection(URL, username,
password);
            //        title_txt, author_txt, publication_txt, price_txt;
            Statement stmt = conn.createStatement();

            String deleteQuery = "DELETE FROM book_info WHERE id = "+ n +"" ;
            stmt.execute(deleteQuery);

            JOptionPane.showMessageDialog(frame, "Record Deleted Succesfully
successfully....");

            selectRecord();
        } catch (Exception e) {
            System.out.println("" + e);
        }
    }

    public void updateRecord() {
        JTextField idField = new JTextField();
        JTextField titleField = new JTextField();
        JTextField authorField = new JTextField();
        JTextField pubField = new JTextField();
        JTextField priceField = new JTextField();

        Object[] fields = {
            "ID:", idField,
            "Title:", titleField,
            "Author:", authorField,
            "Publication:", pubField,
            "Price:", priceField
        };

        int option = JOptionPane.showConfirmDialog(frame, fields, "Update
Record", JOptionPane.OK_CANCEL_OPTION);
        if (option == JOptionPane.OK_OPTION) {
            String id = idField.getText();
            String titleD = titleField.getText();
            String authorD = authorField.getText();
            String pubD = pubField.getText();
            int priceD = Integer.parseInt(priceField.getText());
```

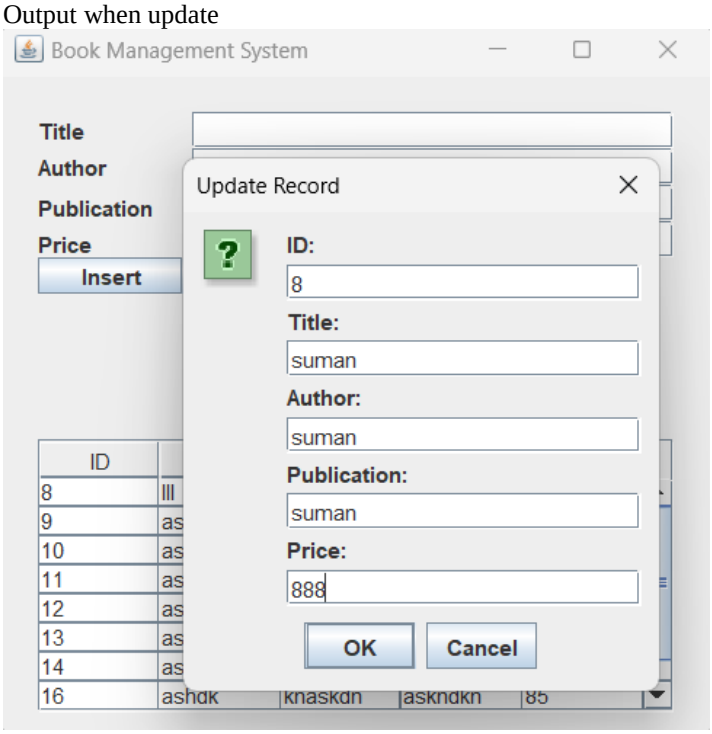
```
            try {
                Class.forName("com.mysql.jdbc.Driver");
                String URL = "jdbc:mysql://localhost:3306/bookdb";
                String username = "root";
                String password = "";
                Connection conn = DriverManager.getConnection(URL, username,
password);
                Statement stmt = conn.createStatement();

                if (!id.isEmpty() && !titleD.isEmpty() && !authorD.isEmpty() &&
!pubD.isEmpty() && priceD > 0) {
                    String updateQuery = "UPDATE book_info SET title = '" +
titleD + "', author = '" + authorD + "', publication = '" + pubD + "', price = '"
+ priceD + "' WHERE id = " + id;
                    stmt.execute(updateQuery);
                    JOptionPane.showMessageDialog(frame, "Record updated
successfully.");
                    selectRecord();
                } else {
                    JOptionPane.showMessageDialog(frame, "Empty TextField");
                    selectRecord();
                }

            } catch (Exception e) {
                e.printStackTrace();
                JOptionPane.showMessageDialog(frame, "Error: " + e.getMessage());
            }
        }
    }

    private void selectRecord() {
        try {
            /*Step 2: Load and Register the MySQL Driver */
            Class.forName("com.mysql.jdbc.Driver");
            /*Step 3: Create and Establish the Connection with MySQL Server*/
            String URL = "jdbc:mysql://localhost:3306/bookdb";
            String username = "root";
            String password = "";
            Connection conn = DriverManager.getConnection(URL, username,
password);

            /*Step 4: Create a Statement */
            Statement stmt = conn.createStatement();

            /*Step 5: Create and Execute SQL statment using above stmt */
```

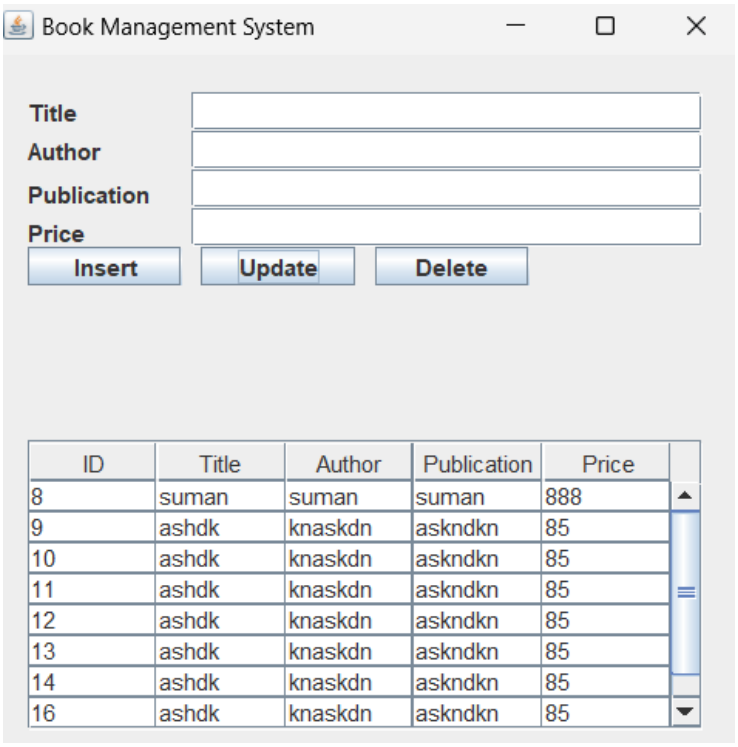
```
            String selectQuery = "Select * from  book_info";
            ResultSet rs = stmt.executeQuery(selectQuery);
            /* Step 6: Check or Process the above query is executed or not */
            tblModel.setRowCount(0);
            while (rs.next()) {
                int id = rs.getInt(1);  //id can be replaced by 1
                String title = rs.getString(2);
                String author = rs.getString(3);
                String pub = rs.getString(4);
                int price = rs.getInt(5);

                Object[] tblData = {id, title, author, pub, price};
                tblModel.addRow(tblData);

            }
            /* Step 7: Close the connection */
            conn.close();

        } catch (ClassNotFoundException | SQLException e) {
            e.printStackTrace();
        }
    }

}

public class BookEntryForm {

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

}
```

Output : when program is run and insert form and display database list

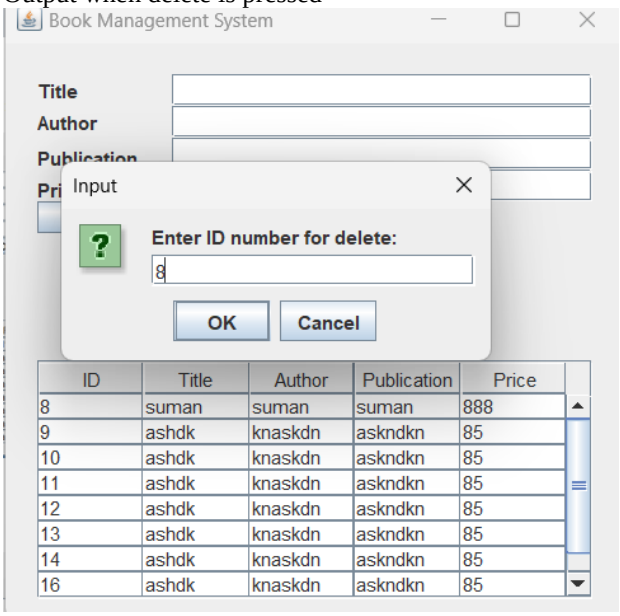



Output when update






Output when delete is pressed



***l. Create a simple login form and enter the username and password in text fields and then click on the Login button, if login success then open the above Book entry form otherwise display incorrect username and password using Swing and JDBC.***

loginForm.java

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package bookentryform;

import java.awt.*;
import java.sql.*;
import javax.swing.*;

class LoginForm
{
    /* Declaration of Object */
   JFrame frame;   /* Declaring object of JFrame.. So, that we can create a frame */
   JLabel lblUser, lblPass;
   JTextField txtUser;
   JPasswordField jpfPass;
   JButton btnLogin, btnReset;

   public LoginForm()
   {

      /* Creating Object of JFrame */
      frame = new JFrame();
      frame.setTitle("Login Form");
      frame.setSize(200, 170);
      frame.setLocationRelativeTo(null); /* Take frame to the Center Screen */
      frame.setLayout(new FlowLayout(FlowLayout.LEFT));
     // frame.setLayout(new GridLayout(3,2));

      /* Creating and Adding Components */
      lblUser = new JLabel("Username: ");
      lblPass = new JLabel("Password: ");
      txtUser = new JTextField(15);
      jpfPass = new JPasswordField(15);
      btnLogin = new JButton("Login");
      btnReset = new JButton("Reset");

      frame.add(lblUser); frame.add(txtUser);
      frame.add(lblPass); frame.add(jpfPass);
      frame.add(btnLogin); frame.add(btnReset);

      btnLogin.addActionListener(e -> loginCheck());
      btnReset.addActionListener(r -> clearFields());

      /* Make frame visible */
      frame.setVisible(true);
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
   }

   private void loginCheck()
   {
```

```
        String user = txtUser.getText();
        String pass = String.valueOf(jpfPass.getPassword());


     try {
          /*Step 2: Load and Register the MySQL Driver */
        Class.forName("com.mysql.jdbc.Driver");
        /*Step 3: Create and Establish the Connection with MySQL Server*/
        String URL = "jdbc:mysql://localhost:3306/bookdb";
        String username="root";
        String password="";
        Connection conn = DriverManager.getConnection(URL, username, password);
        /*Step 4: Create a Statement */
        Statement stmt = conn.createStatement();
        /*Step 5: Create and Execute SQL statment using above stmt */
        String selectQuery = "Select * from  logintbl Where username='"+user+"' and password='"+pass+"'";
        ResultSet rs = stmt.executeQuery(selectQuery);
        /* Step 6: Check or Process the above query is executed or not */
         if(rs.next())
         {
            clearFields();
            JOptionPane.showMessageDialog(frame, "Welcome "+user);
            frame.dispose();
            BEF obj =  new BEF();

         }
         else
         {
            JOptionPane.showMessageDialog(frame, "Username or password do not match...");
         }
        /* Step 7: Close the connection */
        conn.close();
        }
        catch (ClassNotFoundException | SQLException e)
        {
           e.printStackTrace();
        }
     }
    private void clearFields()
    {
        txtUser.setText("");
        jpfPass.setText("");
    }
  }
public class LoginFormBWIC {
    public static void main(String[] args) {
        new LoginForm();
    }
}
```

BookEntryForm.java

```
package bookentryform;
```

```
import java.sql.*;
import javax.swing.*;
import javax.swing.table.*;


class BEF {

    JFrame frame;
    JLabel title, author, publication, price;
    JTextField title_txt, author_txt, publication_txt, price_txt;
    JButton update, insert, delete, view;

     JTable table;
    JScrollPane sp;
    DefaultTableModel tblModel;
    BEF() {
        frame = new JFrame();
        frame.setSize(405, 400);
        frame.setLayout(null);
        frame.setTitle("Book Management System");
        frame.setResizable(true);

        title = new JLabel("Title");
        title.setBounds(20, 20, 80, 20);
        frame.add(title);

        title_txt = new JTextField();
        title_txt.setBounds(105, 20, 265, 20);
        frame.add(title_txt);

        author = new JLabel("Author");
        author.setBounds(20, 40, 80, 20);
        frame.add(author);

        author_txt = new JTextField();
        author_txt.setBounds(105, 40, 265, 20);
        frame.add(author_txt);

        publication = new JLabel("Publication");
        publication.setBounds(20, 60, 80, 25);
        frame.add(publication);

        publication_txt = new JTextField();
        publication_txt.setBounds(105, 60, 265, 20);
        frame.add(publication_txt);

        price = new JLabel("Price");
        price.setBounds(20, 80, 80, 25);
        frame.add(price);

        price_txt = new JTextField();
        price_txt.setBounds(105, 80, 265, 20);
        frame.add(price_txt);

        insert = new JButton("Insert");
        insert.setBounds(20, 100, 80, 20);
```

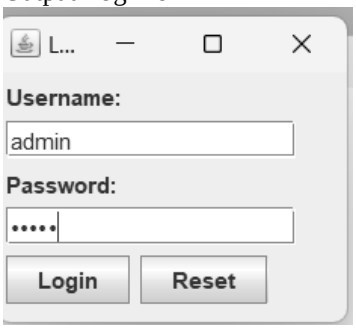
```
        frame.add(insert);

        update = new JButton("Update");
        update.setBounds(110, 100, 80, 20);
        frame.add(update);

        delete = new JButton("Delete");
        delete.setBounds(200, 100, 80, 20);
        frame.add(delete);

        view = new JButton("View");
        view.setBounds(290, 100, 80, 20);
//        frame.add(view);

        /* Creating JTable to add row from database table */
        String[] columns = {"ID","Title","Author","Publication","Price"};

        tblModel = new DefaultTableModel(columns,0);
        table = new JTable(tblModel);
        sp = new JScrollPane(table);
        frame.add(sp);
        sp.setBounds(20, 200, 350, 150);

              selectRecord();/*calling function recordSelect when program is run*/
         insert.addActionListener(e -> insertRecord());
         delete.addActionListener(e -> deleteRecord());
         update.addActionListener(e -> updateRecord());
         view.addActionListener(e -> selectRecord());


        frame.setVisible(true);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);



    }
        public void insertRecord() {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            String URL = "jdbc:mysql://localhost:3306/bookdb";
            String username = "root";
            String password = "";
            Connection conn = DriverManager.getConnection(URL, username, password);
            Statement stmt = conn.createStatement();

            String titleD = title_txt.getText();
            String authorD = author_txt.getText();
            String pubD = publication_txt.getText();
            int priceD = Integer.parseInt(price_txt.getText());
            String priceStr = Integer.toString(priceD);

            if (!titleD.isEmpty() && !authorD.isEmpty() && !pubD.isEmpty() && !priceStr.isEmpty()) {
                String insertQuery = "INSERT INTO book_info (title, author, publication, price) VALUES ('" + titleD +
"', '" + authorD + "', '" + pubD + "', '" + priceStr + "')";
                stmt.execute(insertQuery);
                JOptionPane.showMessageDialog(frame, "Record inserted successfully.");
```

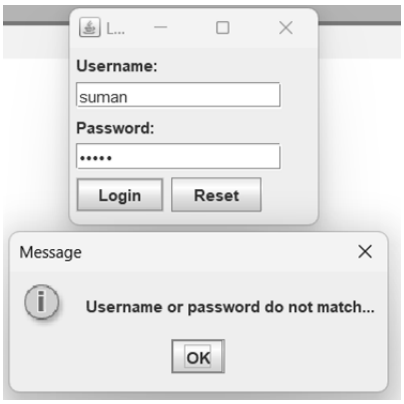
```
            selectRecord();
        } else {
            JOptionPane.showMessageDialog(frame, "Empty TextField");
            selectRecord();
        }

    } catch (Exception e) {
        JOptionPane.showMessageDialog(frame, "Error: " + e.getMessage());
    }
}

    public void deleteRecord(){
        String n = JOptionPane.showInputDialog("Enter ID number for delete:");

            try{
                Class.forName("com.mysql.jdbc.Driver");
         String URL = "jdbc:mysql://localhost:3306/bookdb";
        String username="root";
        String password="";
        Connection conn = DriverManager.getConnection(URL, username, password);
//        title_txt, author_txt, publication_txt, price_txt;
         Statement stmt = conn.createStatement();


       String deleteQuery = "DELETE FROM book_info WHERE id = "+n;
        stmt.execute(deleteQuery);

        JOptionPane.showMessageDialog(frame,"Record Deleted Succesfully successfully....");

        selectRecord();
         }
         catch(Exception e){
            System.out.println(""+e);
         }
        }

    public void updateRecord() {
       JTextField idField = new JTextField();
       JTextField titleField = new JTextField();
       JTextField authorField = new JTextField();
       JTextField pubField = new JTextField();
       JTextField priceField = new JTextField();

       Object[] fields = {
          "ID:", idField,
          "Title:", titleField,
          "Author:", authorField,
          "Publication:", pubField,
          "Price:", priceField
       };

       int option = JOptionPane.showConfirmDialog(frame, fields, "Update Record",
JOptionPane.OK_CANCEL_OPTION);
       if (option == JOptionPane.OK_OPTION) {
          String id = idField.getText();
          String titleD = titleField.getText();
```

```
            String authorD = authorField.getText();
            String pubD = pubField.getText();
            int priceD = Integer.parseInt(priceField.getText());
            String priceStr = Integer.toString(priceD);

            try {
                Class.forName("com.mysql.jdbc.Driver");
                String URL = "jdbc:mysql://localhost:3306/bookdb";
                String username = "root";
                String password = "";
                                    Connection conn = DriverManager.getConnection(URL, username, password);
                Statement stmt = conn.createStatement();

                if (!id.isEmpty() && !titleD.isEmpty() && !authorD.isEmpty() && !pubD.isEmpty() &&
!priceStr.isEmpty()) {
 String updateQuery = "UPDATE book_info SET title = '" + titleD + "', author = '" + authorD + "', publication = '" +
pubD + "', price = '" + priceStr + "' WHERE id = " + id;
                    stmt.execute(updateQuery);
                    JOptionPane.showMessageDialog(frame, "Record updated successfully.");
                    selectRecord();
                } else {
                    JOptionPane.showMessageDialog(frame, "Empty TextField");
                    selectRecord();
                }

            } catch (Exception e) {
                e.printStackTrace();
                JOptionPane.showMessageDialog(frame, "Error: " + e.getMessage());
            }
        }
    }

        private void selectRecord()
    {
        try {
            /*Step 2: Load and Register the MySQL Driver */
        Class.forName("com.mysql.jdbc.Driver");
        /*Step 3: Create and Establish the Connection with MySQL Server*/
        String URL = "jdbc:mysql://localhost:3306/bookdb";
        String username="root";
        String password="";
        Connection conn = DriverManager.getConnection(URL, username, password);

        /*Step 4: Create a Statement */
        Statement stmt = conn.createStatement();

        /*Step 5: Create and Execute SQL statment using above stmt */
        String selectQuery = "Select * from  book_info";
        ResultSet rs = stmt.executeQuery(selectQuery);
        /* Step 6: Check or Process the above query is executed or not */
         tblModel.setRowCount(0);
         while(rs.next())
         {
            int id = rs.getInt(1);  //id can be replaced by 1
            String title = rs.getString(2);
            String author = rs.getString(3);
```

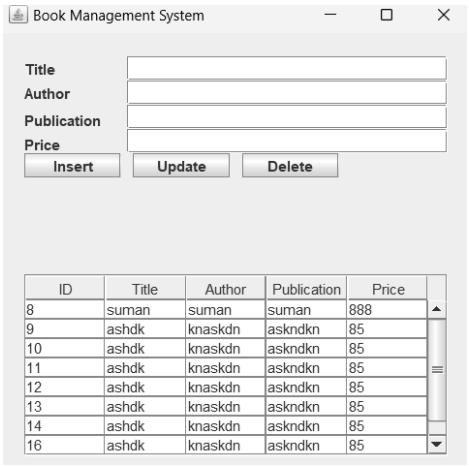
```
            String pub = rs.getString(4);
            int price = rs.getInt(5);

            Object[] tblData = {id, title,author,pub,price};
            tblModel.addRow(tblData);

        }
        /* Step 7: Close the connection */
        conn.close();
        }
        catch (ClassNotFoundException | SQLException e)
        {
           e.printStackTrace();
        }
    }

}
public class BookEntryForm {

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

Output Loginform



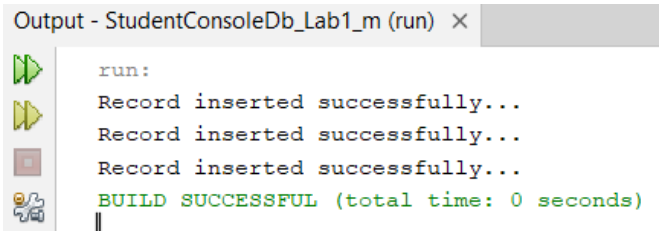

Output bookentryform


Book Management System
Title
Author
Publication
Price
Insert
Update
Delete
ID
Title
Author
Publication
Price
8 suman suman suman 888
9 ashdk knaskdn askndkn 85
10 ashdk knaskdn askndkn 85
11 ashdk knaskdn askndkn 85
12 ashdk knaskdn askndkn 85
13 ashdk knaskdn askndkn 85
14 ashdk knaskdn askndkn 85
16 ashdk knaskdn askndkn 85

***m. Write a Java program that create the database College along with student table that contains the Column Roll no, Name, Level, Division and Major. After that add five records in it using JDBC***

```
package studentconsoledb_lab1_m;

import java.sql.*;

public class StudentConsoleDb_Lab1_m {

    public static void main(String[] args) throws SQLException {

        String URL = "jdbc:mysql://localhost:3306/college";
        String username = "root";
        String password = "";
        Connection conn = DriverManager.getConnection(URL, username, password);

        String insertQuery = "INSERT INTO student VALUES (?, ?, ?, ?, ?)";
        PreparedStatement pstmt = conn.prepareStatement(insertQuery);

        // Insert first record
        pstmt.setInt(1, 101);
        pstmt.setString(2, "Ram");
        pstmt.setString(3, "Bachelor");
        pstmt.setString(4, "First");
        pstmt.setString(5, "Science");
        int rowAffected1 = pstmt.executeUpdate();
        if(rowAffected1 > 0)
            System.out.println("Record inserted successfully...");
        else
            System.out.println("Error while inserting record...");

        // Insert second record
        pstmt.setInt(1, 102);
        pstmt.setString(2, "Sita");
        pstmt.setString(3, "Bachelor");
        pstmt.setString(4, "Distinction");
        pstmt.setString(5, "Data Science");
        int rowAffected2 = pstmt.executeUpdate();
        if(rowAffected2 > 0)
            System.out.println("Record inserted successfully...");
        else
            System.out.println("Error while inserting record...");

        // Insert third record
        pstmt.setInt(1, 103);
        pstmt.setString(2, "Gita");
        pstmt.setString(3, "Bachelor");
        pstmt.setString(4, "Distinction");
        pstmt.setString(5, "Data Science");
        int rowAffected3 = pstmt.executeUpdate();
        if(rowAffected3 > 0)
            System.out.println("Record inserted successfully...");
        else
            System.out.println("Error while inserting record...");

        // Close the connection
```

```
        conn.close();
    }
}
```




| | | id | name | level | divison | najor |
|---|---|---|---|---|---|---|
| ☐ | Edit Copy Delete | 101 | Ram | Bachelor | First | Science |
| ☐ | Edit Copy Delete | 102 | Sita | Bachelor | Distinction | Data Science |
| ☐ | Edit Copy Delete | 103 | Gita | Bachelor | Distinction | Data Science |

***n. Write a program to display the record from the table students who have got distinction and have a major subject Data Science. Assume that the student table in the database College and contains the column Roll no, Name, Level, Division and Major.***

```
        /*New JDBC Mysql Driver Steps */

package displaydistinctionholderanddatasciencestudents_lab1_n;
import java.sql.*; //step 1
public class DisplayDistinctionHolderAndDataScienceStudents_lab1_n {

   public static void main(String[] args) {
     try {
         /*Step 2: Load and Register the MySQL Driver */

      /*Step 3: Create and Establish the Connection with MySQL Server*/
      String URL = "jdbc:mysql://localhost:3306/college";
      String username="root";
      String password="";
      Connection conn = DriverManager.getConnection(URL, username, password);
      /*Step 4: Create a Statement */
      String selectQuery = "Select * from student Where division=? and major=?";
      PreparedStatement pstmt = conn.prepareStatement(selectQuery);
      /*Step 5: Execute SQL Statment */
       pstmt.setString(1, "Distinction");
       pstmt.setString(2, "Data Science");

       ResultSet rs = pstmt.executeQuery();
       System.out.println("Roll\tName\tLevel\t\tDivsion\t\tMajor");
       while(rs.next())
       {
          int roll = rs.getInt(1);
          String name = rs.getString(2);
          String level = rs.getString(3);
          String div = rs.getString(4);
          String major = rs.getString(5);
          System.out.println(roll+"\t"+name+"\t"+level+"\t"+div+"\t"+major);
       }
      /*Step 7: Close the Connection*/
      conn.close();
      }
      catch(Exception e)
      {
         System.out.println(e.getMessage());
      }
   }
 }
```

Output:

```
run:
Roll    Name    Level           Divsion         Major
102     Sita    Bachelor        Distinction     Data Science
103     Gita    Bachelor        Distinction     Data Science
BUILD SUCCESSFUL (total time: 0 seconds)
```