**Practical No – 01**

**OOPs concepts in Java – 1**

1. **Write a program to create a class and implement a default, overloaded and copy Constructor**

**Code :**

class Person

{

String name;

int age;

//default constructor

Person()

{

name="-";

age=0;

}

//parameterized constructor

Person(String name, int age)

{

this.age=age;

this.name=name;

}

//copy constructor

Person(Person p)

{

name=p.name;

age=p.age;

}

void display()

{

System.out.print("\n------------------\nName : "+name+"\nAge : "+age);

}

public static void main(String[] args)

{

Person p1=new Person();

p1.display();

Person p2=new Person("Abhishek", 20);

p2.display();

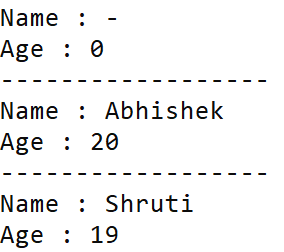
Person p3=new Person("Shruti", 19);

Person p4=new Person(p3);

p4.display();

}

}

**Output :**

1. **Write a program to create a class and implement the concepts of Method Overloading**

**Code :**

class Addition

{

//Method to add two integers

int add(int a, int b)

{

return a+b;

}

//Method to add three integers

int add(int a, int b, int c)

{

return a+b+c;

}

//Method to add two floats

double add(double a, double b)

{

return a+b;

}

//Method to add three floats

double add(double a, double b, double c)

{

return a+b+c;

}

public static void main(String[] args)

{

Addition e1=new Addition();

System.out.println("Sum of (2, 3) : "+e1.add(2, 3));

System.out.println("Sum of (4, 3, 1) : "+e1.add(4, 3, 1));

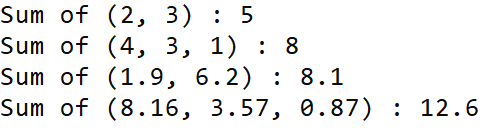
System.out.println("Sum of (1.9, 6.2) : "+e1.add(1.9, 6.2));

System.out.println("Sum of (8.16, 3.57, 0.87) : "+e1.add(8.16, 3.57, 0.87));

}

}

**Output :**

****

1. **Write a program to create a class and implement the concepts of Static methods**

**Code :**

class MathOperations

{

//Static Method to add two values

static int add(int a, int b)

{

return a+b;

}

//Static Method to subtract two values

static int subtract(int a, int b)

{

return a-b;

}

//Static Method to divide two values

static int divide(int a, int b)

{

return a/b;

}

//Static Method to multiply two values

static int multiply(int a, int b)

{

return a\*b;

}

public static void main(String[] args)

{

System.out.println("Addition of (2, 8) : "+MathOperations.add(2, 8));

System.out.println("Subtraction of (12, 5) : "+MathOperations.subtract(12, 5));

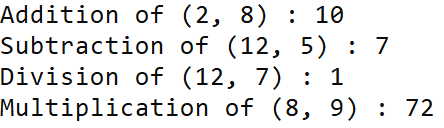
System.out.println("Division of (12, 7) : "+MathOperations.divide(12, 7));

System.out.println("Multiplication of (8, 9) : "+MathOperations.multiply(8, 9));

}

}

**Output :**

****

**Practical No – 02**

**OOPs concepts in Java – 2**

1. **Write a program to implement the concepts of Inheritance and Method overriding**

**Code :**

interface Transaction

{

public void deposit(float ammount);

public void withdraw(float ammount);

}

interface InterestEarning

{

public void applyInterest(float interestRate);

}

class Bank implements Transaction, InterestEarning

{

String customerName;

int accountNumber;

float balance;

Bank(String customerName, int accountNumber,float balance)

{

this.customerName=customerName;

this.accountNumber=accountNumber;

this.balance=balance;

}

@Override

public void deposit(float ammount)

{

balance=balance+ammount;

System.out.println("Remaining Balance after deposit: "+balance);

}

@Override

public void withdraw(float ammount)

{

if(ammount<=balance)

{

balance=balance-ammount;

System.out.println("Remaining Balance after withdrawal: "+balance);

}

else

{

System.out.println("Insufficient Balance!!");

}

}

@Override

public void applyInterest(float interestRate)

{

balance+=balance\*interestRate/100;

System.out.println("Remaining Balance after applying interest : "+balance);

}

void display()

{

System.out.println("Name : "+customerName);

System.out.println("Account Number : "+accountNumber);

System.out.println("Balance : "+balance);

}

}

class Main{

public static void main(String[] args)

{

Bank B=new Bank("Abhishek",1228007867,12450);

B.display();

B.deposit(500);

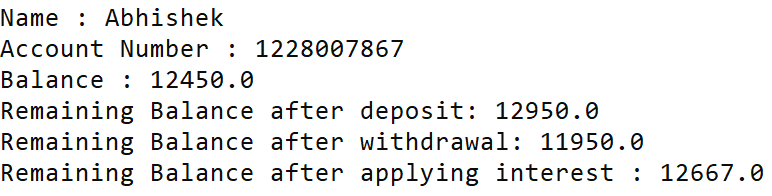
B.withdraw(1000);

B.applyInterest(6);

}

}

**Output :**

****

1. **Write a program to implement the concepts of Abstract classes and methods**

**Code :**

abstract class Employee

{

int empId;

String name;

//Parameterized constructor

Employee(int empId, String name)

{

this.empId=empId;

this.name=name;

}

//Abstract method to calculate salary

abstract float calculateSalary();

void display()

{

System.out.println("-------------------\nName : "+name);

System.out.println("Employee ID : "+empId);

}

}

class SalariedEmployee extends Employee

{

float monthlySalary;

//Parameterized constructor

SalariedEmployee(int empId, String name, float monthlySalary)

{

super(empId, name);

this.monthlySalary=monthlySalary;

}

float calculateSalary()

{

return monthlySalary\*12;

}

}

class HourlyEmployee extends Employee

{

int work, rate=500;

//Parameterized constructor

HourlyEmployee(int empId, String name, int work)

{

super(empId, name);

this.work=work;

}

float calculateSalary()

{

return work\*rate;

}

}

class Main

{

public static void main(String[] args)

{

SalariedEmployee S=new SalariedEmployee(1223, "Abhishek", 48000);

S.display();

System.out.println("Yearly Salary : "+S.calculateSalary());

HourlyEmployee H=new HourlyEmployee(1224, "Niyojana", 45);

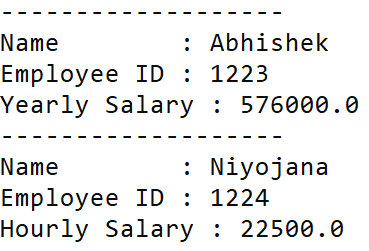
H.display();

System.out.println("Hourly Salary : "+H.calculateSalary());

}

}

**Output :**

****

1. **Write a program to implement the concept of interfaces**

**Code :**

interface Shape

{

//Abstract area method

public double area();

//Abstract perimeter method

public double perimeter();

}

class Circle implements Shape

{

double radius;

//Parameterized constructor

Circle(double radius)

{

this.radius=radius;

}

//Overriden methods from Shape class

@Override

public double area()

{

return Math.PI\*radius\*radius;

}

@Override

public double perimeter()

{

return 2\*Math.PI\*radius;

}

}

class Triangle implements Shape

{

double base, height, side1, side2, side3;

//Parameterized constructor

Triangle(double base, double height)

{

this.base=base;

this.height=height;

}

Triangle(double side1, double side2, double side3)

{

this.side1=side1;

this.side2=side2;

this.side3=side3;

}

//Overriden methods from Shape class

@Override

public double area()

{

return 0.5\*base\*height;

}

@Override

public double perimeter()

{

return side1+side2+side3;

}

}

class Main

{

public static void main(String[] args)

{

Shape circle=new Circle(5.4);

Shape triangle1=new Triangle(4.2, 3.9);

Shape triangle2=new Triangle(12.6, 7.6, 2.9);

System.out.println("Circle Area : "+circle.area());

System.out.println("Circle Perimeter : "+circle.perimeter());

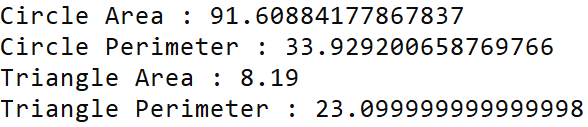
System.out.println("Triangle Area : "+triangle1.area());

System.out.println("Triangle Perimeter : "+triangle2.perimeter());

}

}

**Output :**

****

**Practical No – 03**

**Exceptions**

1. **Write a program to raise built-in exceptions and raise them as per the requirements**

**Code :**

class BuiltInException{

    void arithmeticException(int num1, int num2){

        try{

            System.out.println("Division of "+num1+" & "+num2+" is "+num1/num2);

        }catch(ArithmeticException e){

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

        }

    }

    void arrayIndexOutOfBoundsException(int[] ary, int searchIndex){

        try{

            System.out.println("Value at index "+searchIndex+" is "+ary[searchIndex]);

        }catch(ArrayIndexOutOfBoundsException e){

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

        }

    }

     void nullPointerException(String str){

        try{

            System.out.println("Length of string is "+str.length());

        }catch(NullPointerException e){

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

        }

    }

}

class Main{

    public static void main(String[] args){

        BuiltInException e1=new BuiltInException();

        int[] ary={2, 3, 5, 7, 11};

        int searchIndex=2;

        String str="SRM COLLEGE";

        System.out.println("Normal flow..");

        e1.arithmeticException(10, 5);

        e1.arrayIndexOutOfBoundsException(ary, searchIndex);

        e1.nullPointerException(str);

        System.out.println("\nException cases..");

        searchIndex=10;

        str=null;

        e1.arithmeticException(10, 0);

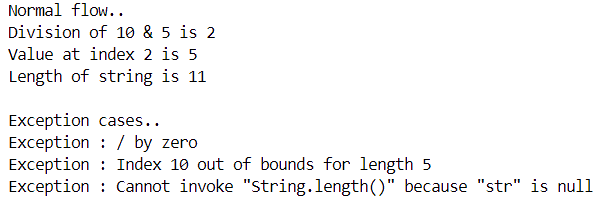
        e1.arrayIndexOutOfBoundsException(ary, searchIndex);

        e1.nullPointerException(str);

    }

}

**Output :**

****

1. **Write a program to define user defined exceptions and raise them as per the requirements**

**Code :**

import java.util.\*;

class AuthenticationException extends Exception

{

AuthenticationException(String message)

{

super(message);

}

}

class IsUserValid

{

public static void main(String[] args)

{

Scanner scan=new Scanner(System.in);

String Username, Password;

System.out.print("Username : ");

Username=scan.nextLine();

System.out.print("Password : ");

Password=scan.nextLine();

try

{

if(Username.equals("admin") && Password.equals("admin"))

{

System.out.println("Valid User!");

}

else

{

throw new AuthenticationException("Invalid User!");

}

}

catch(AuthenticationException e)

{

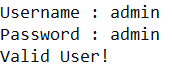
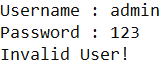
System.out.println(e.getMessage());

}

}

}

**Output :**

**** ****

**Practical No – 06**

**Swing**

1. **Create a swing application that randomly changes color on button click**

**Code :**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.util.Random;

class P6A extends JFrame implements ActionListener{

JButton btn;

JPanel panel;

Random r=new Random();

P6A(){

setSize(500, 500);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLayout(new FlowLayout());

btn=new JButton("Change color");

btn.setFocusPainted(false);

panel=new JPanel();

add(panel, BorderLayout.CENTER);

panel.setPreferredSize(new Dimension(500, 500));

panel.add(btn, BorderLayout.CENTER);

btn.addActionListener(this);

setVisible(true);

}

public void actionPerformed(ActionEvent e){

panel.setBackground(new Color(r.nextInt(255), r.nextInt(255), r.nextInt(255)));

}

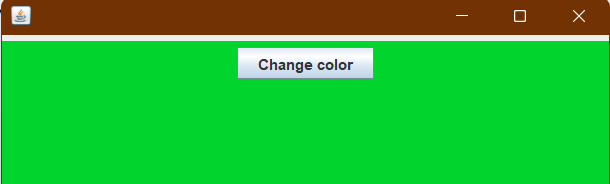
public static void main(String[] args){

new P6A();

}

}

**Output :**



1. **Create a Swing application to demonstrate use of TextArea using scrollpane to show contest of text file in textarea selected using file chooser**

**Code :**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.io.\*;

import java.util.\*;

class P6B extends JFrame implements ActionListener{

JTextArea textArea;

JScrollPane scrollPane;

JFileChooser fileChooser;

JButton openBtn;

P6B(){

setSize(800, 800);

setResizable(false);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLayout(new FlowLayout());

openBtn=new JButton("Open File");

add(openBtn);

textArea=new JTextArea(40, 70);

add(textArea);

scrollPane=new JScrollPane(textArea);

add(scrollPane);

scrollPane.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL\_SCROLLBAR\_ALWAYS);

scrollPane.setVerticalScrollBarPolicy(JScrollPane.VERTICAL\_SCROLLBAR\_ALWAYS);

openBtn.addActionListener(this);

setVisible(true);

}

public void actionPerformed(ActionEvent e){

// If open button is clicked

if(e.getSource()==openBtn){

fileChooser=new JFileChooser();

// Open file chooser dialogbox

int i=fileChooser.showOpenDialog(this);

// If file is selected and open button is clicked

if(i==JFileChooser.APPROVE\_OPTION){

textArea.setText("");

File filePath=fileChooser.getSelectedFile();

try{

Scanner sc=new Scanner(filePath);

// Read file line by line and append to text area and exception handling is compulsory

while(sc.hasNextLine()){

textArea.append(sc.nextLine()+"\n");

}

sc.close();

}catch(Exception ex){

System.out.println(e.toString());

}

}

}

}

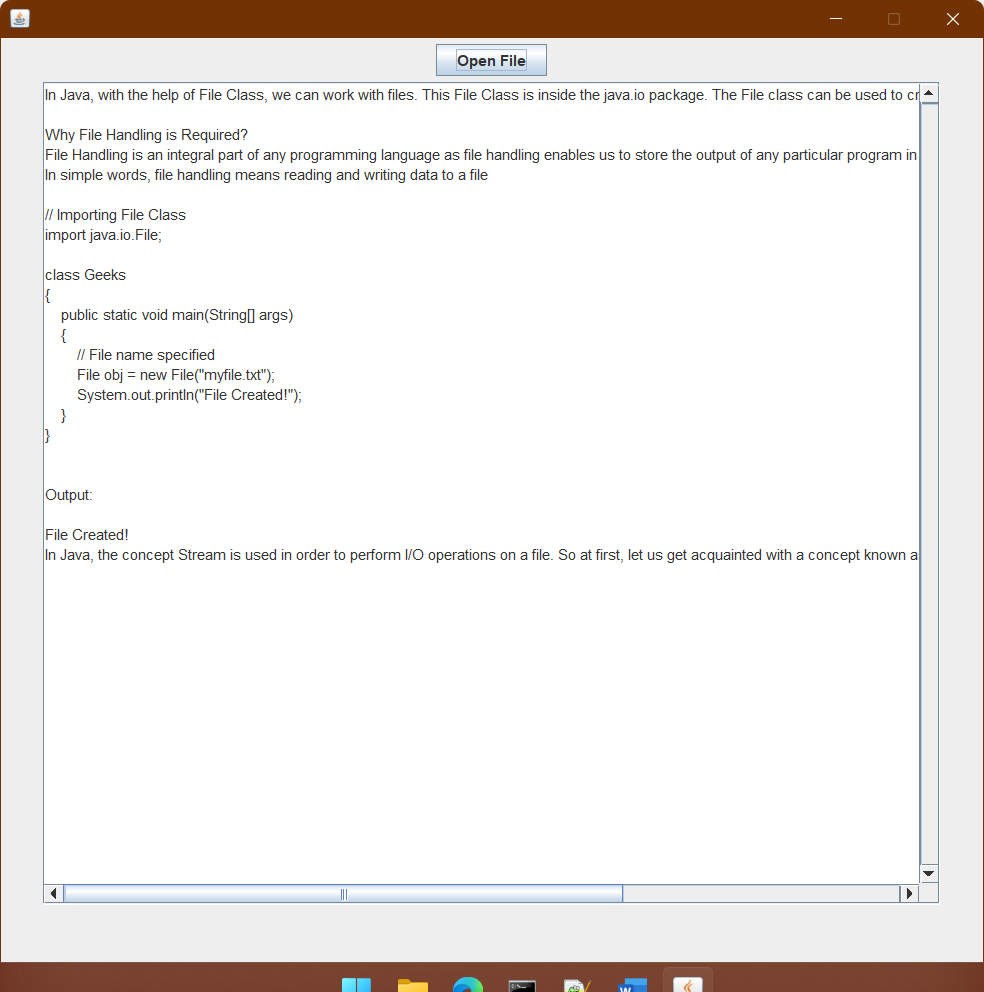
public static void main(String[] args){

new P6B();

}

}

**Output :**

****

1. **Create a Swing application to demonstrate use of scrollpane to change its color selected using colour chooser**

**Code :**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

class P6C extends JFrame implements ActionListener{

JButton chooseBtn;

JPanel mainPanel;

P6C(){

// Set frame properties -->

setSize(600, 600);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setResizable(false);

setLayout(new BorderLayout());

// Adding Components -->

mainPanel=new JPanel();

chooseBtn=new JButton("Choose Color");

// Adding components to frame -->

add(mainPanel, BorderLayout.CENTER);

add(chooseBtn, BorderLayout.NORTH);

chooseBtn.addActionListener(this);

setVisible(true);

}

public void actionPerformed(ActionEvent e){

if(e.getSource()==chooseBtn){

Color color=JColorChooser.showDialog(this, "Choose Color", Color.WHITE);

mainPanel.setBackground(color);

}

}

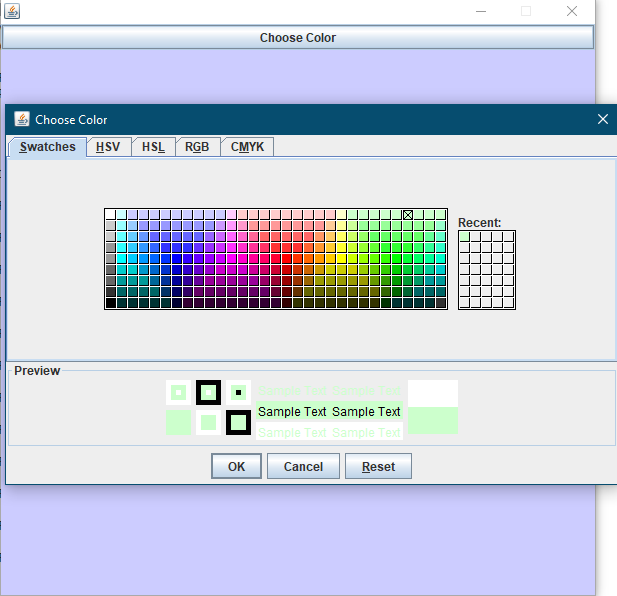
public static void main(String[] args){

new P6C();

}

}

**Output :**

****

**Practical No – 07**

**Layouts**

1. **Flow Layout**

**Code :**

import javax.swing.\*;

import java.awt.event.\*;

import java.awt.\*;

class Main extends JFrame implements ActionListener{

static String username="admin", password="admin";

JTextField un, pwd;

JButton btnLogin;

JLabel loginState;

Main(){

// Frame properties --->

setSize(600, 600);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLayout(new FlowLayout());

// Adding components --->

un=new JTextField(10);

pwd=new JTextField(10);

btnLogin=new JButton("Login");

loginState=new JLabel();

btnLogin.setFocusPainted(false);

add(un);

add(pwd);

add(btnLogin);

add(loginState);

btnLogin.addActionListener(this);

setVisible(true);

}

@Override

public void actionPerformed(ActionEvent e){

// Checking if login button is clicked or not --->

if(e.getSource()==btnLogin){

String u=un.getText();

String p=pwd.getText();

if(u.equals(username) && p.equals(password)){

loginState.setText("Login Successfull...");

}else{

loginState.setText("Login Unsuccessfull...");

}

}

}

public static void main(String[] args){

new Main();

}

}

**Output :**

****

1. **Grid Layout**

**Code :**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

class P7B extends JFrame implements ActionListener{

String[] symbols={"C", "%", "+", "B",

"7", "8", "9", "-",

"4", "5", "6", "x",

"1", "2", "3", "÷",

"00", "0", ".", "="};

JPanel outputPanel, btnPanel;

JButton[] btns=new JButton[20];

JTextField outputField;

Font outputFont=new Font("calibri", Font.PLAIN, 30);

Font defaultFont=new Font("calibri", Font.PLAIN, 15);

P7B(){

// Putting the default font for entire application component -->

UIManager.put("Button.font", defaultFont);

UIManager.put("Label.font", defaultFont);

// Setting up frame -->

setSize(400,400);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setResizable(false);

setLayout(new BorderLayout());

// Component Config -->

outputPanel=new JPanel();

btnPanel=new JPanel(new GridLayout(5, 4, 5, 5));

outputField=new JTextField();

outputField.setPreferredSize(new Dimension(370, 60));

outputField.setFont(outputFont);

add(outputPanel, BorderLayout.NORTH);

add(btnPanel, BorderLayout.CENTER);

add(btnPanel, BorderLayout.CENTER);

outputPanel.add(outputField);

// Adding buttons to the panel

for(int i=0; i<20; i++){

btns[i]=new JButton(symbols[i]);

btnPanel.add(btns[i]);

btns[i].setFocusPainted(false);

btns[i].addActionListener(this);

}

setVisible(true);

}

public void actionPerformed(ActionEvent e){

// Type converted of getSource to JButton -->

JButton clickedBtn=(JButton)e.getSource();

// Getting the symbol of the button which is clicked -\_>

String clickedSymbol=clickedBtn.getText();

// Storing the output field value -->

String expression=outputField.getText();

double operand1=0.0, operand2=0.0;

String operator="";

if(clickedSymbol=="C"){

expression="";

outputField.setText("");

}else if(clickedSymbol=="B"){

// Deleting the last element by creating a substring from 0 to length-1

expression=expression.substring(0, expression.length()-1);

outputField.setText(expression);

}else if(clickedSymbol=="="){

int index=0;

if(expression.contains("+")){

operator="+";

index=expression.indexOf("+");

}else if(expression.contains("-")){

operator="-";

index=expression.indexOf("-");

}else if(expression.contains("x")){

operator="\*";

index=expression.indexOf("x");

}else if(expression.contains("÷")){

operator="/";

index=expression.indexOf("÷");

}else if(expression.contains("%")){

operator="%";

index=expression.indexOf("%");

}

operand1=Integer.parseInt(expression.substring(0, index));

operand2=Integer.parseInt(expression.substring(index+1, expression.length()));

if(operator=="+"){

outputField.setText(Double.toString(operand1+operand2));

}else if(operator=="-"){

outputField.setText(Double.toString(operand1-operand2));

}else if(operator=="\*"){

outputField.setText(Double.toString(operand1\*operand2));

}else if(operator=="/"){

outputField.setText(Double.toString(operand1/operand2));

}else if(operator=="%"){

outputField.setText(Double.toString(operand1%operand2));

}

}else{

expression=outputField.getText();

outputField.setText(expression+clickedSymbol);

}

}

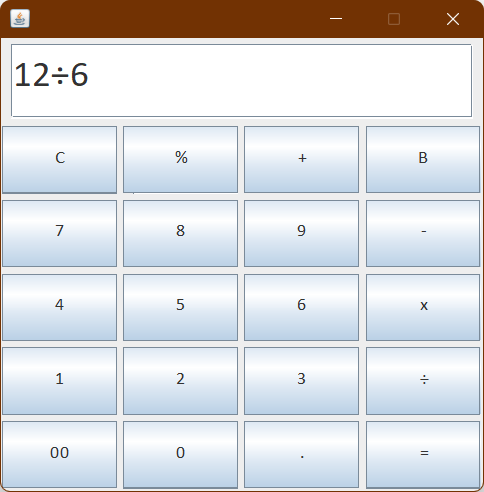
public static void main(String[] args){

new P7B();

}

}

**Output :**

****

1. **Border Layout**

**Code :**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

class P7C extends JFrame implements ActionListener{

JPanel mainPanel;

JLabel label;

JButton nextBtn, prevBtn;

int imgIndex=0;

String[] image={"C:/Users/abhis/Desktop/SEM 4/Java/images/img1.jpg",

"C:/Users/abhis/Desktop/SEM 4/Java/images/img2.jpg",

"C:/Users/abhis/Desktop/SEM 4/Java/images/img3.jpg"};

P7C(){

// Setting frame properties -->

setSize(900, 600);

setResizable(false);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLayout(new BorderLayout());

// Creating components -->

mainPanel=new JPanel(new BorderLayout());

label=new JLabel(new ImageIcon(image[imgIndex]));

nextBtn=new JButton("Next");

prevBtn=new JButton("Previous");

// Adding components -->

add(mainPanel, BorderLayout.CENTER);

mainPanel.add(label, BorderLayout.CENTER);

mainPanel.add(prevBtn, BorderLayout.WEST);

mainPanel.add(nextBtn, BorderLayout.EAST);

prevBtn.addActionListener(this);

nextBtn.addActionListener(this);

setVisible(true);

}

public void actionPerformed(ActionEvent e){

if(e.getSource()==prevBtn){

imgIndex=((imgIndex-1)+image.length)%image.length;

}else if(e.getSource()==nextBtn){

imgIndex=(imgIndex+1)%image.length;

}

label.setIcon(new ImageIcon(image[imgIndex]));

}

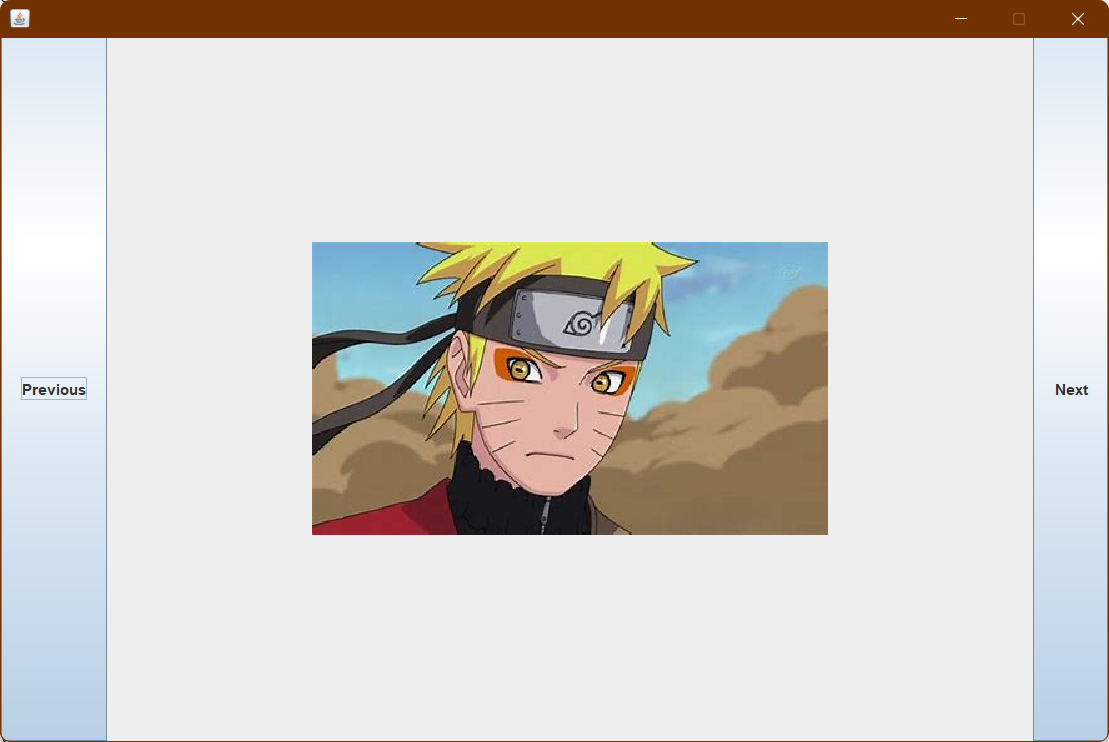
public static void main(String[] args){

new P7C();

}

}

**Output :**

****

**Practical No – 08**

**Events**

1. **ActionEvent**

**Code :**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

class P8A extends JFrame implements ActionListener{

JLabel label, ansLabel;

JTextField inputField;

JButton cubeBtn;

P8A(){

setSize(400, 400);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLayout(new FlowLayout());

label=new JLabel("Enter a number");

inputField=new JTextField(10);

cubeBtn=new JButton("Cube");

ansLabel=new JLabel();

add(label);

add(inputField);

add(cubeBtn);

add(ansLabel);

cubeBtn.addActionListener(this);

setVisible(true);

}

public void actionPerformed(ActionEvent e){

int number=Integer.parseInt(inputField.getText());

ansLabel.setText("Cube is : "+Integer.toString(number\*number\*number));

}

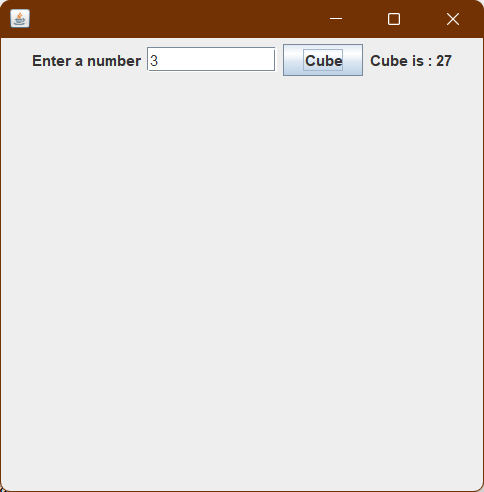
public static void main(String[] args){

new P8A();

}

}

**Output :**

****

1. **MouseEvent**

**Code :**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.util.Random;

class P8B extends JFrame implements MouseListener{

JLabel stateLabel;

JPanel panel;

Random r=new Random();

P8B(){

// Setting frame properties -->

setSize(400, 400);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLayout(new BorderLayout());

// Configuring components -->

stateLabel=new JLabel("Mouse Events");

panel=new JPanel();

add(stateLabel, BorderLayout.NORTH);

add(panel, BorderLayout.CENTER);

panel.setBackground(new Color(255, 255, 255));

setVisible(true);

panel.addMouseListener(this);

}

public void mouseReleased(MouseEvent e){

panel.setBackground(new Color(r.nextInt(255), r.nextInt(255), r.nextInt(255)));

stateLabel.setText("Mouse released");

}

public void mouseClicked(MouseEvent e){

panel.setBackground(new Color(r.nextInt(255), r.nextInt(255), r.nextInt(255)));

stateLabel.setText("Mouse clicked");

}

public void mouseExited(MouseEvent e){

panel.setBackground(new Color(r.nextInt(255), r.nextInt(255), r.nextInt(255)));

stateLabel.setText("Mouse exited");

}

public void mouseEntered(MouseEvent e){

panel.setBackground(new Color(r.nextInt(255), r.nextInt(255), r.nextInt(255)));

stateLabel.setText("Mouse entered");

}

public void mousePressed(MouseEvent e){

panel.setBackground(new Color(r.nextInt(255), r.nextInt(255), r.nextInt(255)));

stateLabel.setText("Mouse pressed");

}

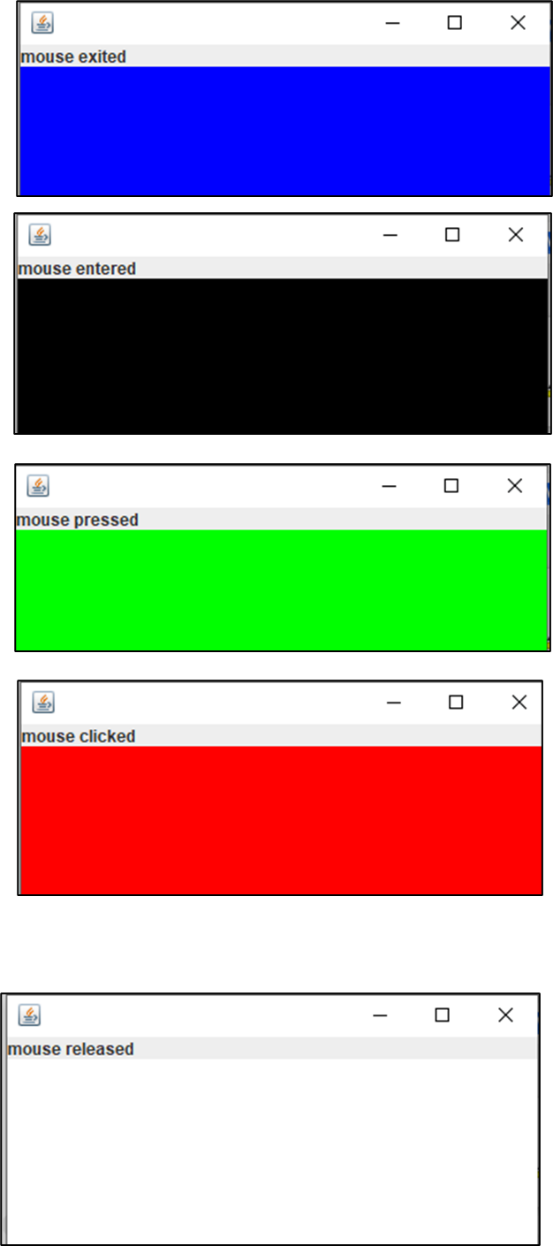
public static void main(String[] args){

new P8B();

}

}

****

**Output :**

1. **KeyEvent**

**Code :**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

class P8C extends JFrame implements KeyListener{

JLabel state;

JTextField inputField;

P8C(){

setSize(600, 600);

setResizable(false);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLayout(new FlowLayout());

inputField=new JTextField(10);

state=new JLabel();

add(inputField);

add(state);

inputField.addKeyListener(this);

setVisible(true);

}

public void keyPressed(KeyEvent event1){

state.setText("Enter string...");

}

public void keyTyped(KeyEvent event2){

state.setText("typing...");

}

public void keyReleased(KeyEvent event3){

// inputField.getText().trim().split(" ").length --> counts the number of words in field

state.setText("Word count : "+(inputField.getText().trim().split(" ").length));

}

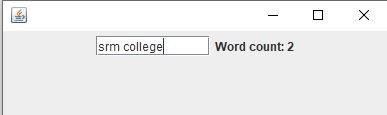
public static void main(String[] args){

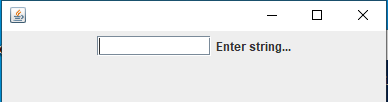
new P8C();

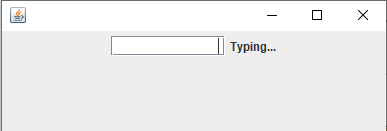
}

}

**Output :**







1. **SelectionEvent**

**Code :**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.event.\*;

class P8D extends JFrame implements ListSelectionListener{

JLabel l;

JList<String> list;

String[] lang={"Python", "Java", "C", "CPP"};

String[] info={

"Python is a high-level interpreted language",

"Java is a high-level programming language",

"C is an efficient, all-purpose programming language",

"C++ is an object-oriented programming language"};

P8D(){

setSize(400, 400);

setLayout(new FlowLayout());

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

l=new JLabel("Select language : ");

list=new JList<String>(lang);

add(l);

add(list);

list.addListSelectionListener(this);

setVisible(true);

}

public void valueChanged(ListSelectionEvent e){

if(!e.getValueIsAdjusting()){

int index=list.getSelectedIndex();

if(index >= 0){

JOptionPane.showMessageDialog(null, info[index]);

}

}

}

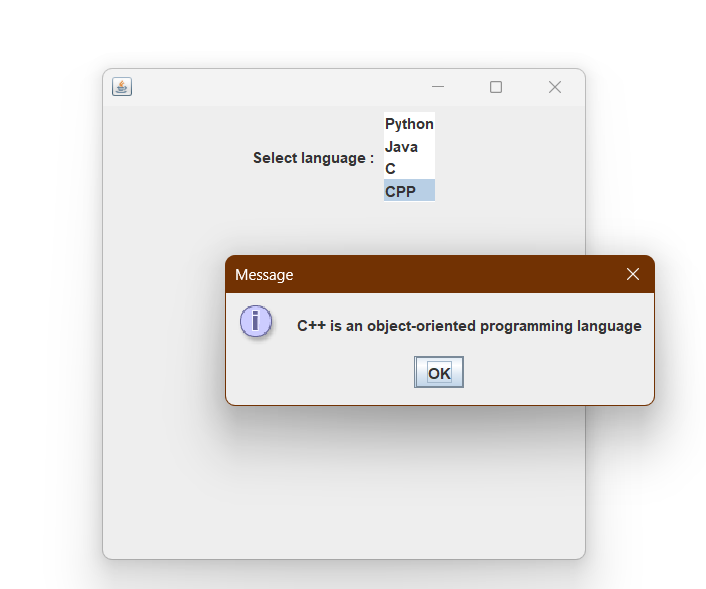
public static void main(String[] args){

new P8D();

}

}

**Output :**

****

1. **FocusEvent**

**Code :**

import javax.swing.\*;

import java.awt.event.\*;

import java.awt.\*;

class P8E extends JFrame implements FocusListener{

JTextField usernameField, passwordField;

JButton submitBtn;

JLabel usernameLabel, passwordLabel, warningLabel;

P8E(){

// Setting frame properties -->

setSize(300, 400);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLayout(new FlowLayout());

// Creating components and adding them

usernameLabel=new JLabel("Username");

usernameField=new JTextField(10);

passwordLabel=new JLabel("Password");

passwordField=new JTextField(10);

submitBtn=new JButton("Submit");

warningLabel=new JLabel();

add(usernameLabel);

add(usernameField);

add(passwordLabel);

add(passwordField);

add(submitBtn);

add(warningLabel);

// Adding listeners

usernameField.addFocusListener(this);

setVisible(true);

}

public void focusLost(FocusEvent e){

if(e.getSource()==usernameField){

if(usernameField.getText().trim().isEmpty()){

warningLabel.setText("First fill Username");

}

}

}

public void focusGained(FocusEvent e){

warningLabel.setText("");

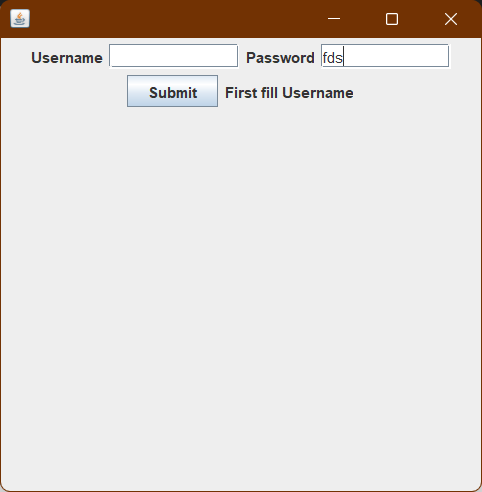
}

public static void main(String[] args){

new P8E();

}

}

**Output :**

**Code :**

import java.awt.event.\*;

import java.awt.\*;

import javax.swing.\*;

class P8Bb extends JFrame implements MouseMotionListener{

JLabel l;

Graphics g;

P8Bb(){

setSize(400, 500);

setLayout(new BorderLayout());

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

l=new JLabel(" ");

add(l, BorderLayout.NORTH);

addMouseMotionListener(this);

setVisible(true);

}

public void mouseDragged(MouseEvent e){

g=getGraphics();

g.setColor(Color.RED);

g.fillOval(e.getX(), e.getY(), 20, 20);

}

public void mouseMoved(MouseEvent e){

l.setText("X : "+e.getX()+" Y : "+e.getY());

}

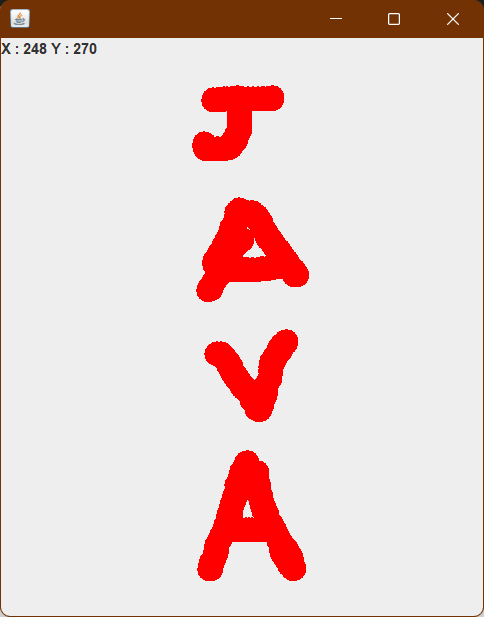
public static void main(String[] args){

new P8Bb();

}

}

**Output :**

****

**Practical No – 05**

**JDBC**

1. **Write a JDBC program that displays the data of a given table in a GUI Table**

**Code :**

import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

import java.awt.\*;

import java.sql.\*;

class P5A{

public static void main(String[] args){

// Setting frame properties

JFrame frame=new JFrame("Database Table Viewer");

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setSize(600, 400);

// Component config

JTable table=new JTable();

JScrollPane scrollPane=new JScrollPane(table);

frame.add(scrollPane, BorderLayout.CENTER);

DefaultTableModel model=new DefaultTableModel();

table.setModel(model);

try{

Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/college", "root", "admin");

Statement stmt=con.createStatement();

ResultSet resultSet=stmt.executeQuery("select \* from employee");

ResultSetMetaData metaData=resultSet.getMetaData();

int columnCount=metaData.getColumnCount();

for(int i=1; i<=columnCount; i++){

model.addColumn(metaData.getColumnName(i));

}

while(resultSet.next()){

Object[] row=new Object[columnCount];

for(int i=1; i<=columnCount; i++){

row[i-1]=resultSet.getObject(i);

}

model.addRow(row);

}

}catch(SQLException e){

JOptionPane.showMessageDialog(frame, "Error: " + e.getMessage(), "Database Error", JOptionPane.ERROR\_MESSAGE);

e.printStackTrace();

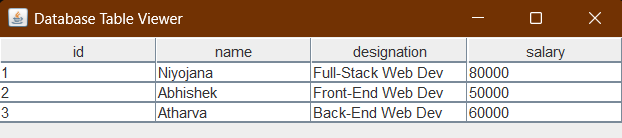
}

frame.setVisible(true);

}

}

**Output :**

****

1. **Write a JDBC program to Show the details of a specified product from a given table selected using Combobox**

**Code :**

import java.sql.\*;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

class P5B extends JFrame implements ActionListener{

JComboBox box;

JLabel label;

JButton retriveBtn;

JTextArea txt;

Connection con;

ResultSet rs;

Statement st;

P5B(){

// Setting frame properties

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setSize(500, 500);

setLayout(new FlowLayout());

setVisible(true);

// Component config

label=new JLabel("Product Details");

add(label, BorderLayout.NORTH);

txt=new JTextArea(10, 20);

add(txt, BorderLayout.CENTER);

box=new JComboBox();

add(box, BorderLayout.CENTER);

retriveBtn=new JButton("Retrieve");

add(retriveBtn, BorderLayout.CENTER);

retriveBtn.addActionListener(this);

// Database queries

try{

con=DriverManager.getConnection("jdbc:mysql://localhost:3306/product", "root", "admin");

Statement st=con.createStatement();

rs=st.executeQuery("select name from prod");

while(rs.next()){

box.addItem(rs.getString("name"));

}

}catch(SQLException e){

e.printStackTrace();

}

}

public void actionPerformed(ActionEvent a){

try{

String val=(String)box.getSelectedItem();

PreparedStatement ps=con.prepareStatement("select \* from prod where name=?");

ps.setString(1, val);

rs=ps.executeQuery();

while(rs.next()){

txt.setText(" ");

txt.append(" ID : " + rs.getInt("id") + "\n");

txt.append(" Name : " + rs.getString("name") + "\n");

txt.append(" Price : " + rs.getInt("price") + "\n");

txt.append(" Quantity : " + rs.getInt("qty") + "\n");

}

}catch(SQLException e){

e.printStackTrace();

}

}

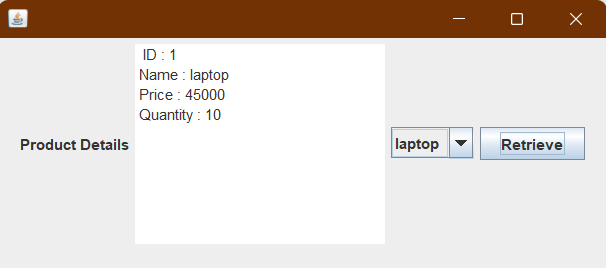
public static void main(String[] args) {

new P5B();

}

}

**Output :**

****

1. **Write a GUI application to Navigate forward and reverse result set data**

**Code :**

import java.sql.\*;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.util.\*;

class P5C extends JFrame implements ActionListener{

JFrame f;

JLabel l;

JPanel p;

JButton nxt, prev;

JTextArea txt;

Connection con;

ResultSet rs;

Statement st;

P5C(){

// Setting the frame properties

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setSize(500, 500);

setLayout(new FlowLayout());

setVisible(true);

// Component config

l=new JLabel("Student Details");

add(l,BorderLayout.NORTH);

txt=new JTextArea(20, 30);

add(txt, BorderLayout.CENTER);

p=new JPanel();

add(p, BorderLayout.SOUTH);

nxt=new JButton("Next");

p.add(nxt, BorderLayout.SOUTH);

prev=new JButton("Previous");

p.add(prev, BorderLayout.SOUTH);

nxt.addActionListener(this);

prev.addActionListener(this);

try{

con=DriverManager.getConnection("jdbc:mysql://localhost:3306/college", "root", "admin");

Statement st=con.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE, ResultSet.CONCUR\_READ\_ONLY);

rs=st.executeQuery("select \* from stud");

if(rs.next()){

show();

}

}catch(SQLException e){

e.getMessage();

}

}

public void show()

{

try{

txt.setText(" ");

txt.append("ID :"+rs.getObject("id")+"\n");

txt.append("Name :"+rs.getObject("name")+"\n");

txt.append("Address :"+rs.getObject("address")+"\n");

txt.append("Class :"+rs.getObject("class")+"\n");

}catch(SQLException e){

e.printStackTrace();

}

}

public void actionPerformed(ActionEvent a){

try{

if(a.getSource()==prev){

if(rs.previous()){

show();

}else{

txt.setText("No previous Records!!");

}

}

if(a.getSource()==nxt){

if(rs.next()){

show();

}else{

txt.setText("No More Records!!");

}

}

}catch(SQLException e){

e.getMessage();

}

}

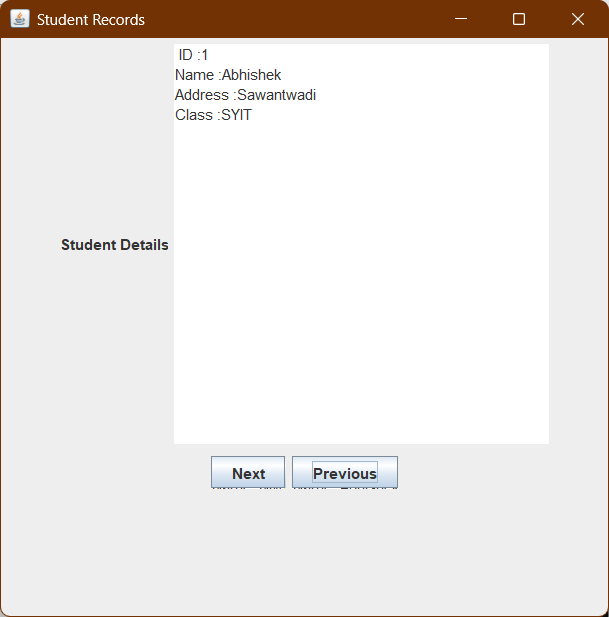
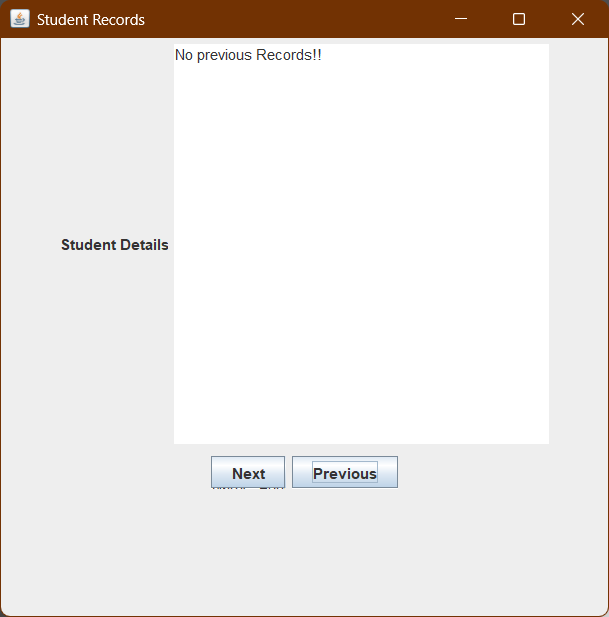
public static void main(String[] args){

new P5C();

}

}

**Output :**

**** ****

**Practical No – 10**

**Demonstrate the use of Anonymous Inner Class in Event Handling**

**Code :**

import javax.swing.\*;

import java.awt.event.\*;

import java.awt.\*;

class P10 extends JPanel{

int x=150; // Initial X position of the square

int y=150; // Initial Y position of the square

JFrame frame;

P10(){

// Setting frame properties

frame=new JFrame("KeyEvent Example");

frame.setSize(400, 300);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

// Add this panel to the frame

frame.add(this);

setFocusable(true); // Make the panel focusable to listen to key events

frame.setVisible(true);

addKeyListener(new KeyAdapter(){

@Override

public void keyPressed(KeyEvent e){

int keyCode=e.getKeyCode();

System.out.println(keyCode);

if(keyCode==KeyEvent.VK\_LEFT){

x-=5;

}

if(keyCode==KeyEvent.VK\_RIGHT){

x+=5;

}

if(keyCode==KeyEvent.VK\_UP){

y-=5;

}

if(keyCode==KeyEvent.VK\_DOWN){

y+=5;

}

// Repaint the panel to reflect the new position of the square

repaint();

}});

}

@Override

protected void paintComponent(Graphics g){

super.paintComponent(g);

// Draw the square at the current (x, y) position

g.setColor(Color.BLUE);

g.fillRect(x, y, 80, 80);

}

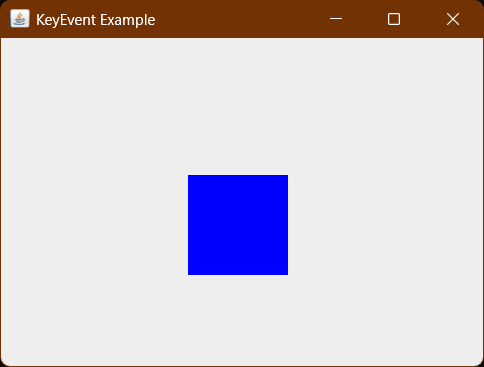
public static void main(String[] args){

new P10();

}

}

**Output :**

****

**Practical No – 09**

**Demonstrate the use of Adapter Class in Event Handling**

**Code :**

import javax.swing.\*;

import java.awt.event.\*;

import java.awt.\*;

class P9{

JFrame frame;

JLabel l1, l2;

JTextField t1, t2;

P9(){

// Setting frame properties

frame=new JFrame("KeyEvent Example");

frame.setSize(400, 300);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

l1=new JLabel("Name : ");

l2=new JLabel("Contact : ");

t1=new JTextField(10);

t2=new JTextField(10);

frame.add(l1);

frame.add(t1);

frame.add(l2);

frame.add(t2);

frame.setLayout(new FlowLayout());

t2.addFocusListener(new FocusAdapter(){ // Anonymous inner class

@Override

public void focusLost(FocusEvent e){

String a=t2.getText().trim();

if(a.length()!=10 || !a.matches("\\d{10}")){

// Invalid mobile number, show a message and highlight the field

JOptionPane.showMessageDialog(frame, "Please enter a valid 10-digit mobile number.", "Invalid Input", JOptionPane.ERROR\_MESSAGE);

}

}

});

frame.setVisible(true);

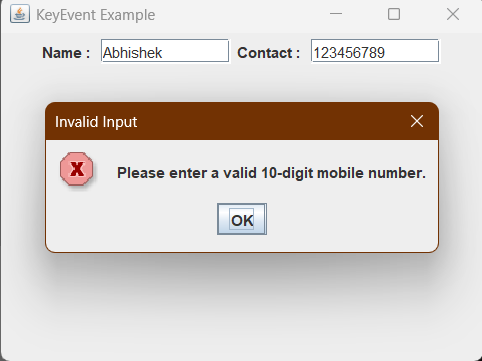
}

public static void main(String[] args){

new P9();

}

}

**Output :**