

Name:- Manish Shashikant Jadhav

Roll no.: - 2201933

PRN no.: - 2030408246006

Subject:- Advanced Java Programming Lab

Experiment no.1

Aim:- Implementing Program that are easily extensible demonstrating Polymorphic behavior.

Program no.1 :- A program using Override method in Java:-

```
package com.AJPExperiments;
class bike{
    public void model()
    {
        System.out.println("We are bikes");
    }
}
class continentalGT extends bike{
    public void model()
    {
        System.out.println("ContinentalGT is model of Royal Enfield");
    }
}
class Dominor extends bike{
    public void model()
    {
        System.out.println("Dominor is model of Bajaj");
    }
}
class NinjaH2R extends bike{
    public void model()
    {
        System.out.println("NinjaH2R is model of Kawasaki");
    }
}

public class Experiment1A {
    public static void main(String[] args) {
        bike m;
        bike m1 = new continentalGT();
        m1.model();
```

```

        bike m2 = new Dominor();
        m2.model();
        bike m3 = new NinjaH2R();
        m3.model();
    }
}

```

Output:-

```

Run: Experiment1A
C:\Program Files\Java\jdk-17.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2\lib\idea_rt.jar=63556:C:\Program Files\JetBrains\I
ContinentalGT is model of Royal Enfield
Dominor is model of Bajaj
NinjaH2R is model of Kawasaki
Process finished with exit code 0

```

Program no.2 :- Write a program to print rate of interest of some banks using overriding method.

```
package com.AJPExperiments;
```

```

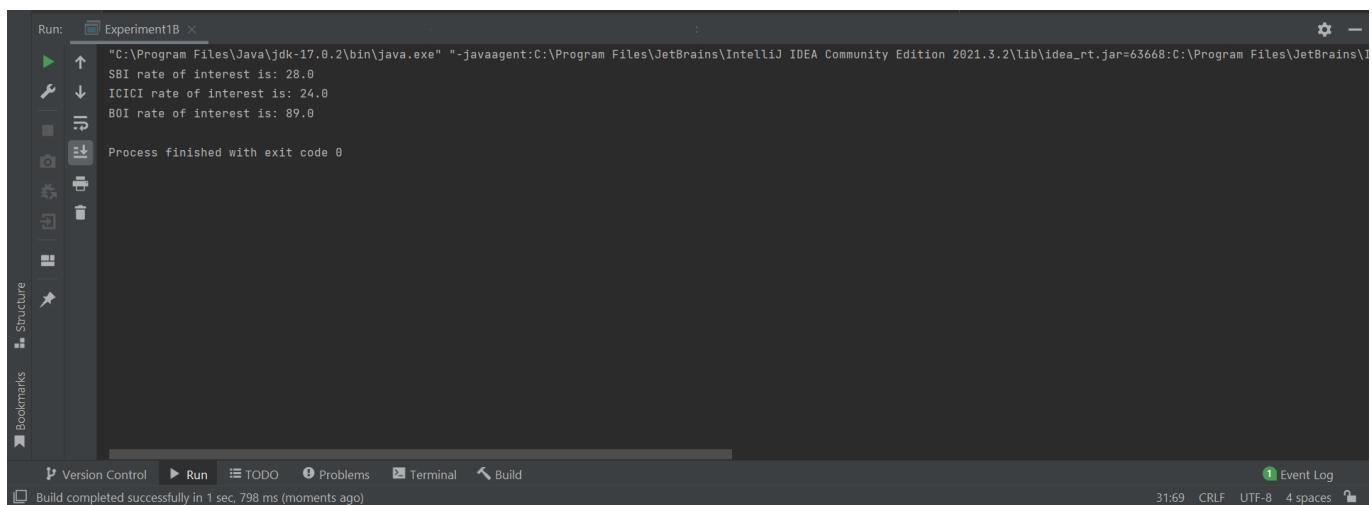
class bank1{
    float interest() {
        return 0;
    }
}
class SBI extends bank1 {
    float interest() {
        return (28.0f);
    }
}
class ICICI extends bank1 {
    float interest() {
        return (24.0f);
    }
}
class BOI extends bank1 {
    float interest() {

```

```
        return (89.0f);
    }
}

public class Experiment1B {
    public static void main(String[] args) {
        bank1 b;
        bank1 b1 = new SBI();
        System.out.println("SBI rate of interest is: "+b1.interest());
        bank1 b2 = new ICICI();
        System.out.println("ICICI rate of interest is: "+b2.interest());
        bank1 b3 = new BOI();
        System.out.println("BOI rate of interest is: "+b3.interest());
    }
}
```

Output:-



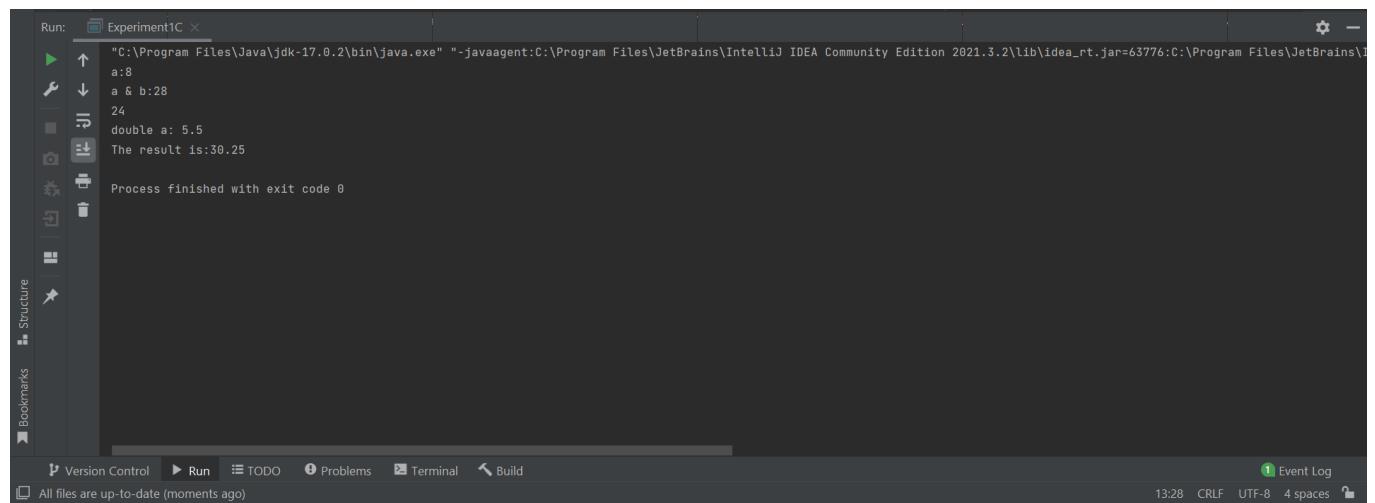
Program no.3 :- Write a program using method overloading.

```
package com.AJPExperiments;
```

```
class overload{
    void demo (int a)
    {
        System.out.println("a:"+a);
    }
    void demo (int a, int b)
    {
        System.out.println("a & b:"+a);
        System.out.println(b);
    }
}
```

```
double demo (double a){  
    System.out.println("double a: "+a);  
    return a*a;  
}  
}  
  
public class Experiment1C {  
    public static void main(String[] args) {  
        overload s1 = new overload();  
        double result;  
        s1.demo(8);  
        s1.demo(28, 24);  
        result = s1.demo(5.5);  
        System.out.println("The result is:"+ result);  
    }  
}
```

Output:-



```
Run: Experiment1C ×  
"C:\Program Files\Java\jdk-17.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2\lib\idea_rt.jar=63776:C:\Program Files\JetBrains\I  
a:8  
a & b:28  
24  
double a: 5.5  
The result is:30.25  
Process finished with exit code 0
```

Name:- Manish Shashikant Jadhav

Roll no.: - 2201933

PRN no.: - 2030408246006

Subject:- Advanced Java Programming Lab

Experiment no.2

Aim:- Implementing an interface.

Program no.1:-

```
package com.AJPExperiments;

public interface Bike {
    public void model();
    public void color();
}

abstract class continentalGt implements Bike{
    public void model()
    {
        System.out.println("ContinentalGT is model of Royal Enfield");
    }
    public void color(){
        System.out.println("ContinentalGt comes with Chrome Finish");
    }
}

class main2{
    public static void main(String args[]){
        continentalGt m = new continentalGt() {};
        m.model();
        m.color();
    }
}
```

Output:-

The screenshot shows the IntelliJ IDEA interface with the 'Run' tab selected. The output window displays the following text:

```
"D:\Program Files\Java\jdk-17.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2\lib\idea_rt.jar=57007:C:\Program Files\JetBrains\IdeaCommunity2021.3\bin\runnerw.jar" -Dfile.encoding=UTF-8 main2
ContinentalGT is model of Royal Enfield
ContinentalGT comes with Chrome Finish

Process finished with exit code 0
```

Below the output window, the navigation bar includes tabs for Version Control, Run, TODO, Problems, Terminal, Build, and Event Log. The Event Log tab is currently active. The status bar at the bottom shows the build time as "Build completed successfully in 1 sec, 774 ms (moments ago)" and the file encoding as "UTF-8".

Program no.2 :-

```
package com.AJPExperiments;

public interface Bikers {
    public void model();
    public void color();
}

abstract class Pulsar implements Bikers{
    public void model(){
        System.out.println("Model of Bajaj");
    }
    public void color(){
        System.out.println("Color is Red and black");
    }
}

class manish{
    public static void main(String args[]){
        Bikers obj = new Pulsar() {};
        obj.model();
        obj.color();
    }
}
```

Output:-

```
"C:\Program Files\Java\jdk-17.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2\lib\idea_rt.jar=57075:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2\bin" -Dfile.encoding=UTF-8 manish
Model of Bajaj
Color is Red and black

Process finished with exit code 0
```

Program no.3 :-

```
package com.AJPExperiments;

public interface biker {
    public void Ninja();
}

interface biker1 extends biker{
    public void Dominor();
}

class demo2 implements biker1{
    public void Ninja(){
        System.out.println("I am NinjaH2R");
    }
    public void Dominor(){
        System.out.println("I am Dominor");
    }
}

class biker3{
    public static void main(String args[]){
        biker b1 = new demo2();
        b1.Ninja();
        biker1 b2 = new demo2();
        b2.Dominor();

    }
}
```

Output:-

```
"C:\Program Files\Java\jdk-17.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2\lib\idea_rt.jar=57100:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2\bin" biker3
I am NinjaH2R
I am Dominor
Process finished with exit code 0
```

Writeup & Oral (4)	Practical Performance (4)	Attendance (2)	Total (10)

Name:- Manish Shashikant Jadhav

Roll no.: 2201933

PRN no.: 2030408246006

Subject:- Advanced Java Programming Lab

Experiment no.3

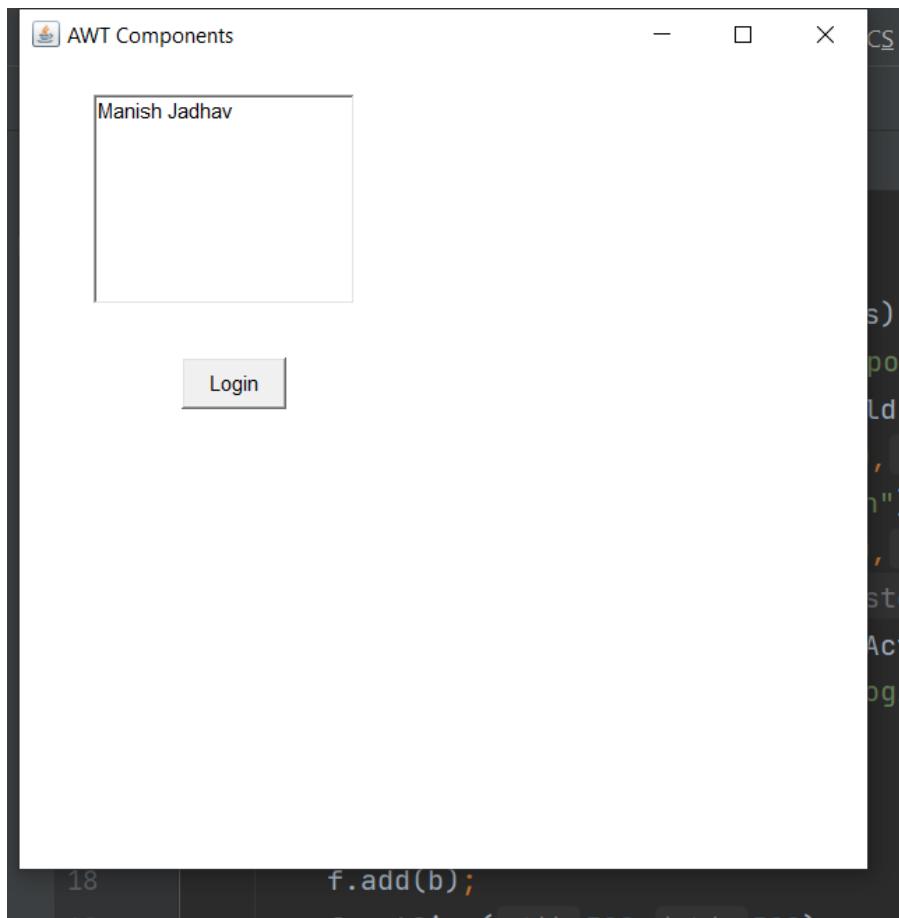
Aim:- Implementing GUI interface using AWT Components.

Program:-

```
package com.AJPExperiments;
import java.awt.*;
import java.awt.event.*;

public class Button3 {
    public static void main(String[] args) {
        Frame f = new Frame("AWT Components");
        final TextField tf = new TextField();
        tf.setBounds(50,50,150,120);
        Button b = new Button("Login");
        b.setBounds(100,200,60,30);
        b.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                tf.setText("Welcome to Login Page");
            }
        });
        f.add(tf);
        f.add(b);
        f.setSize(500,500);
        f.setLayout(null);
        f.setVisible(true);
    }
}
```

Output:-



Writeup & Oral (4)	Practical Performance (4)	Attendance (2)	Total (10)

Name:- Manish Shashikant Jadhav

Roll no.: - 2201933

PRN no.: - 2030408246006

Subject:- Advanced Java Programming Lab

Experiment no.4

Aim:- Implementing GUI interface using various Swing Components.

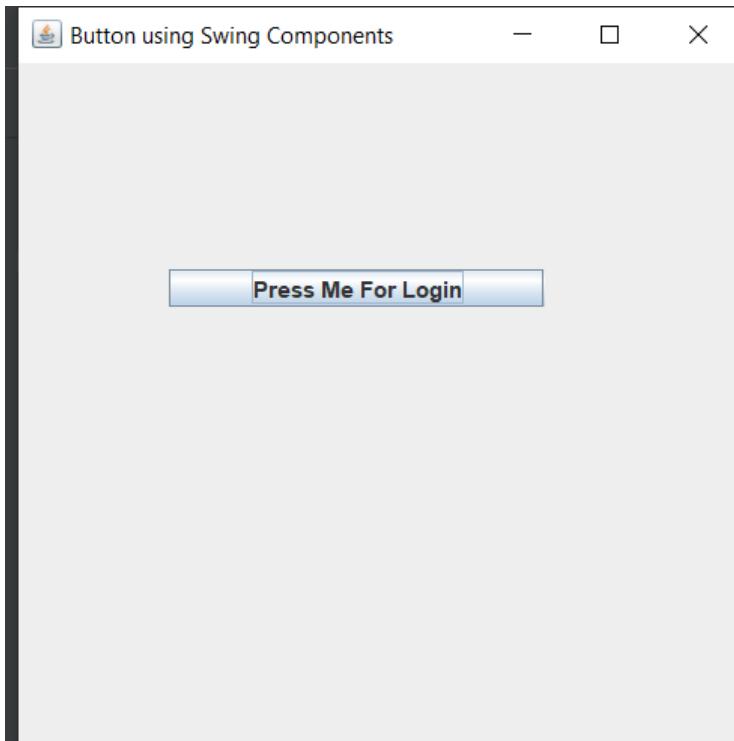
Program A):- Swing Components- JButton.

Input:-

```
package com.AJPExperiments;
import javax.swing.*;

public class Button {
    public static void main(String[] args) {
        JFrame a = new JFrame("Button using Swing Components");
        JButton b = new JButton("Press Me For Login");
        b.setBounds(80,110,200,20);
        a.add(b);
        a.setSize(400,400);
        a.setLayout(null);
        a.setVisible(true);
    }
}
```

Ouput:-



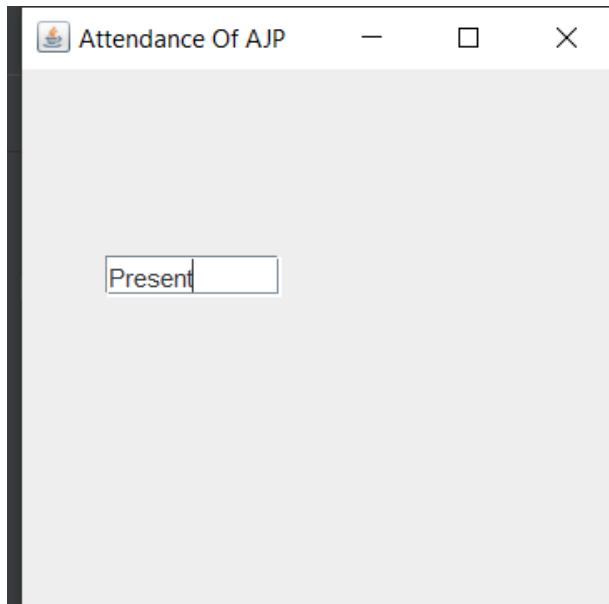
Program B):- Swing Components- JTextField.

Input:-

```
package com.AJPExperiments;
import javax.swing.*;

public class TextField {
    public static void main(String[] args) {
        JFrame a = new JFrame("Attendance Of AJP");
        JTextField t = new JTextField("Present");
        t.setBounds(40,90,85,20);
        a.add(t);
        a.setSize(300,300);
        a.setLayout(null);
        a.setVisible(true);
    }
}
```

Output:-



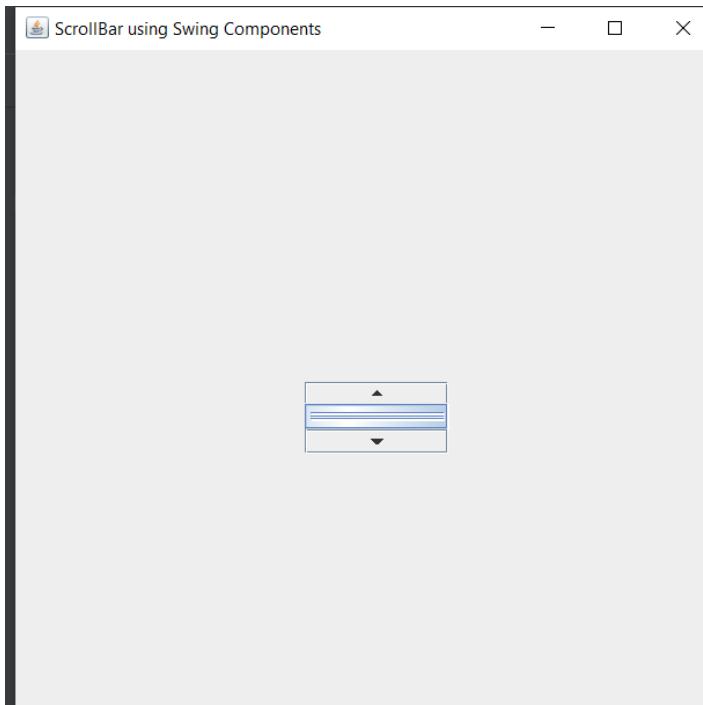
Program C):- Swing Components- JScrollBar.

Input:-

```
package com.AJPExperiments;
import javax.swing.*;

public class ScrollBar {
    public static void main(String[] args) {
        JFrame a = new JFrame("ScrollBar using Swing Components");
        JScrollBar s = new JScrollBar();
        s.setBounds(200,230,100,50);
        a.add(s);
        a.setSize(500,500);
        a.setLayout(null);
        a.setVisible(true);
    }
}
```

Output:-



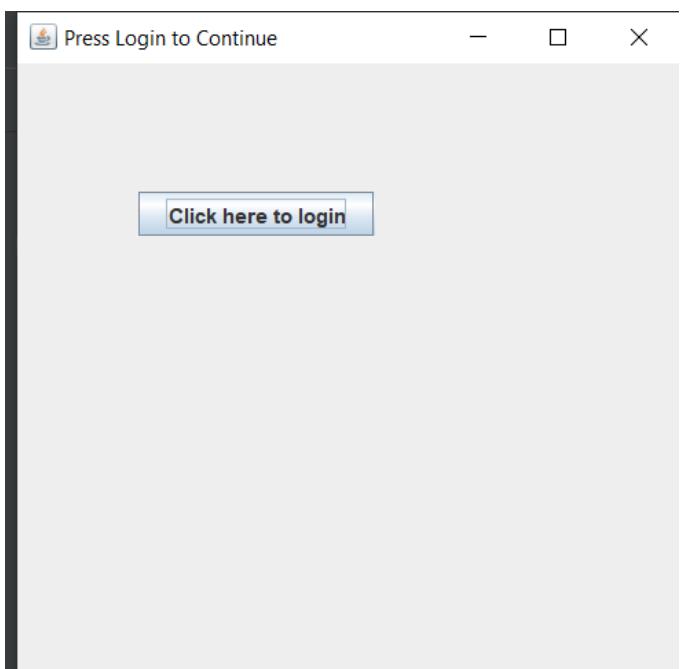
Program C):- Swing Components- JPanel.

Input:-

```
package com.AJPExperiments;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;

public class Panel {
    public static void main(String[] args) {
        JFrame a = new JFrame("Press Login to Continue");
        JPanel p = new JPanel();
        p.setBounds(40,70,200,200);
        JButton b = new JButton("Click here to login");
        b.setBounds(60,50,80,40);
        p.add(b);
        a.add(p);
        a.setSize(400,400);
        a.setLayout(null);
        a.setVisible(true);
    }
}
```

Output:-



Writeup & Oral (4)	Practical Performance (4)	Attendance (2)	Total (10)

Name:- Manish Shashikant Jadhav

Roll no.: 2201933

PRN no.: 2030408246006

Subject:- Advanced Java Programming Lab

Experiment no.5

Aim:- Implementing GUI interface using various Swing Components.

Program A):-Program of Border Layout.

Input:-

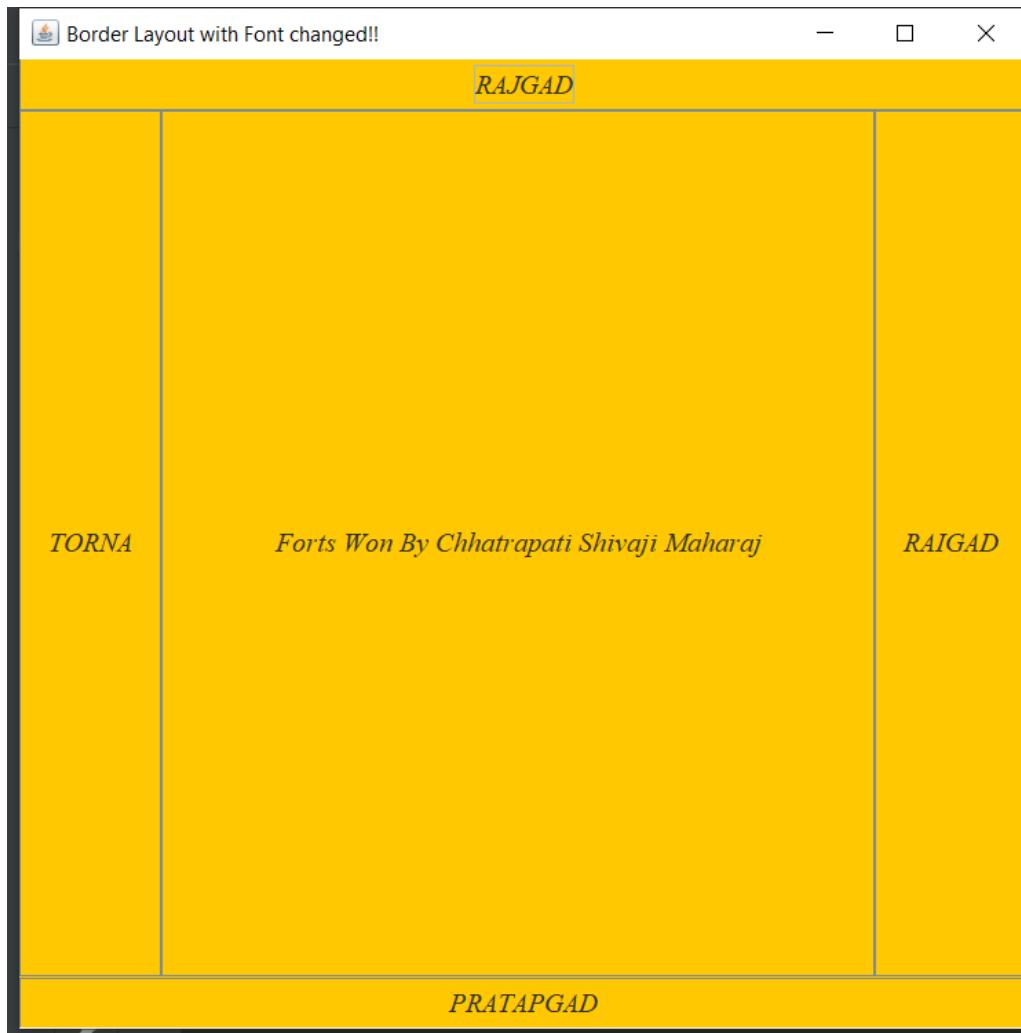
```
package com.AJPExperiments;
import java.awt.*;
import javax.swing.JButton;
import javax.swing.JFrame;

public class Border {
    JFrame f;
    Border(){
        f = new JFrame("Border Layout with Font changed!!");
        JButton b1 = new JButton("RAIGAD");
        JButton b2 = new JButton("TORNA");
        JButton b3 = new JButton("PRATAPGAD");
        JButton b4 = new JButton("RAJGAD");
        JButton b5 = new JButton("Forts Won By Chhatrapati Shivaji Maharaj");
        f.add(b1, BorderLayout.EAST);
        f.add(b2, BorderLayout.WEST);
        f.add(b3, BorderLayout.SOUTH);
        f.add(b4, BorderLayout.NORTH);
        f.add(b5, BorderLayout.CENTER);
        b1.setFont(new Font("Times New Roman",Font.ITALIC,16));
        b2.setFont(new Font("Times New Roman",Font.ITALIC,16));
        b3.setFont(new Font("Times New Roman",Font.ITALIC,16));
        b4.setFont(new Font("Times New Roman",Font.ITALIC,16));
        b5.setFont(new Font("Times New Roman",Font.ITALIC,16));
        b1.setBackground(Color.orange);
        b2.setBackground(Color.orange);
        b3.setBackground(Color.orange);
        b4.setBackground(Color.orange);
        b5.setBackground(Color.orange);
```

```
f.setSize(600,600);
f.setVisible(true);
}

public static void main(String[] args) {
    Border obj = new Border();
}
}
```

Output:-



Program B):-Program of Grid Layout.

Input:-

```
package com.AJPExperiments;
import java.awt.*;
import javax.swing.*;
```

```
public class Grid {  
    JFrame fr;  
    Grid(){  
        fr = new JFrame("Hindu Religious Books");  
        JButton b1 = new JButton("MAHABHARATA");  
        JButton b2 = new JButton("BHAGAVAD GITA");  
        JButton b3 = new JButton("RAMAYANA");  
        JButton b4 = new JButton("ATHARVAVEDA");  
        JButton b5 = new JButton("BHRAHMANA");  
        JButton b6 = new JButton("PANCHATANTRA");  
        JButton b7 = new JButton("UPANISHADAS");  
        JButton b8 = new JButton("PURANAS");  
        JButton b9 = new JButton("ARTHASHASTRA");  
        JButton b10 = new JButton("RIGVEDA");  
        b1.setFont(new Font("Copperplate Gothic Bold",Font.ITALIC,16));  
        b2.setFont(new Font("Copperplate Gothic Bold",Font.ITALIC,16));  
        b3.setFont(new Font("Copperplate Gothic Bold",Font.ITALIC,16));  
        b4.setFont(new Font("Copperplate Gothic Bold",Font.ITALIC,16));  
        b5.setFont(new Font("Copperplate Gothic Bold",Font.ITALIC,16));  
        b6.setFont(new Font("Copperplate Gothic Bold",Font.ITALIC,16));  
        b7.setFont(new Font("Copperplate Gothic Bold",Font.ITALIC,16));  
        b8.setFont(new Font("Copperplate Gothic Bold",Font.ITALIC,16));  
        b9.setFont(new Font("Copperplate Gothic Bold",Font.ITALIC,16));  
        b10.setFont(new Font("Copperplate Gothic Bold",Font.ITALIC,16));  
        b1.setBackground(Color.red);  
        b2.setBackground(Color.red);  
        b3.setBackground(Color.red);  
        b4.setBackground(Color.red);  
        b5.setBackground(Color.red);  
        b6.setBackground(Color.red);  
        b7.setBackground(Color.red);  
        b8.setBackground(Color.red);  
        b9.setBackground(Color.red);  
        b10.setBackground(Color.red);  
        fr.add(b1);  
        fr.add(b2);  
        fr.add(b3);  
        fr.add(b4);  
        fr.add(b5);  
        fr.add(b6);  
        fr.add(b7);  
        fr.add(b8);  
        fr.add(b9);  
        fr.add(b10);  
        fr.setLayout(new GridLayout(2,5));  
        fr.setSize(950,500);  
        fr.setVisible(true);  
    }  
}
```

```

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

```

Output:-



Practical Performance (4)	Writeup & Oral (4)	Attendance (2)	Total (10)

Name:- Manish Shashikant Jadhav

Roll no.: 2201933

PRN no.: 2030408246006

Subject:- Advanced Java Programming Lab

Experiment no.6

Aim:- Implementation of Event Handling using Swing/AWT.

Program A):-

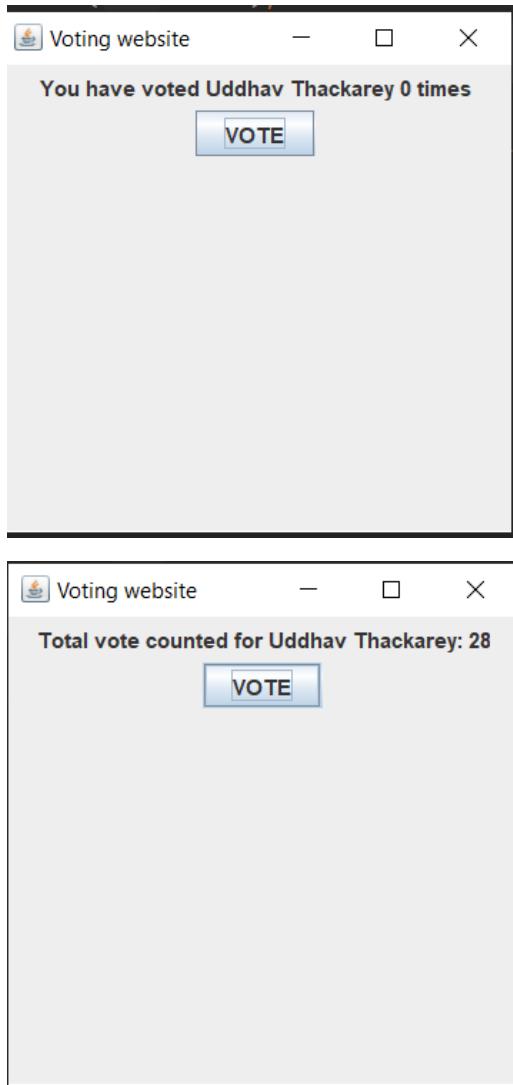
Input:-

```
package com.AJPExperiments;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class event extends JFrame implements ActionListener {
    int count = 0;
    JLabel l1;
    event(){
        setLayout(new FlowLayout());
        l1 = new JLabel("You have voted Uddhav Thackarey 0 times");
        JButton b5 = new JButton("VOTE");
        b5.addActionListener(this);
        add(l1);
        add(b5);
    }
    public void actionPerformed(ActionEvent e){
        count++;
        l1.setText("Total vote counted for Uddhav Thackarey: "+count);
    }

    public static void main(String[] args) {
        event e = new event();
        e.setTitle("Voting website");
        e.setBounds(300,300,300,300);
        e.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        e.setVisible(true);
    }
}
```

Output:-



Program B):-

Input:-

```
package com.AJPExperiments;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class event3 extends JFrame implements ActionListener{
    JRadioButton rb1,rb2,rb3;
```

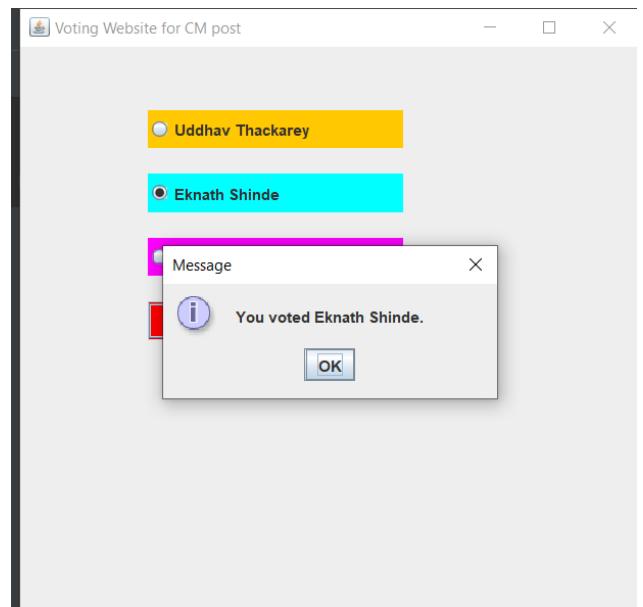
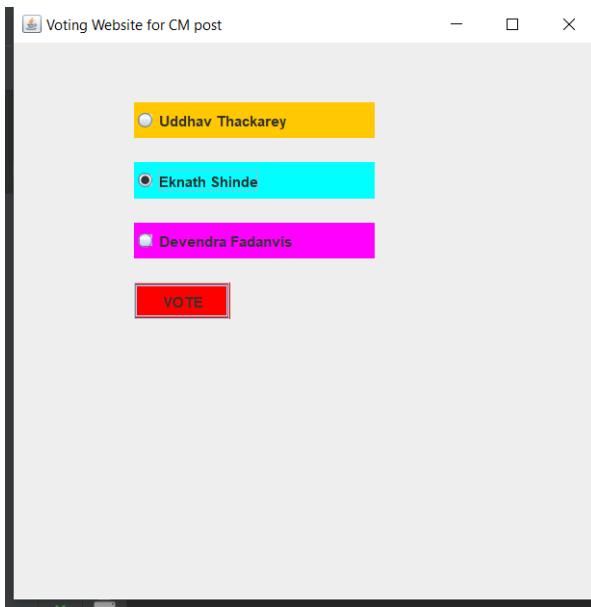
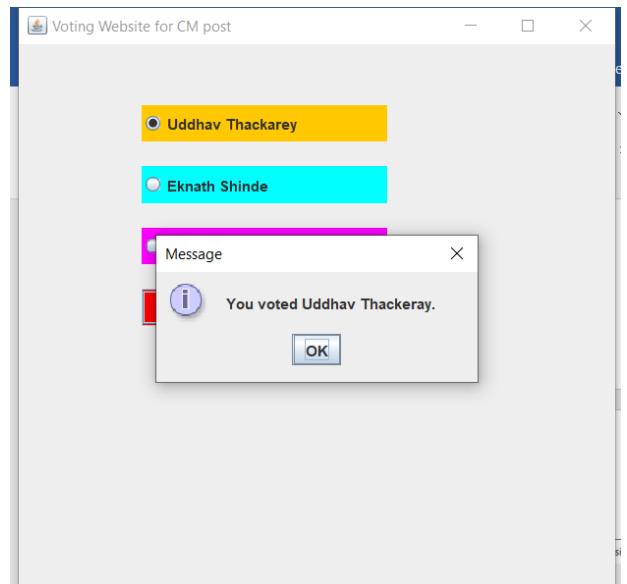
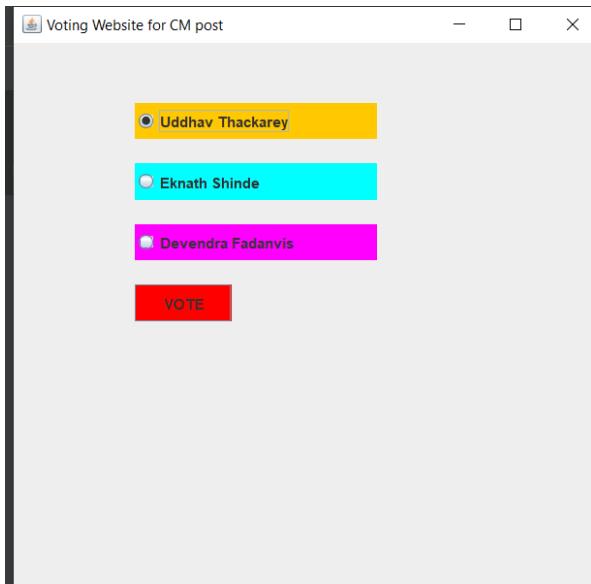
```

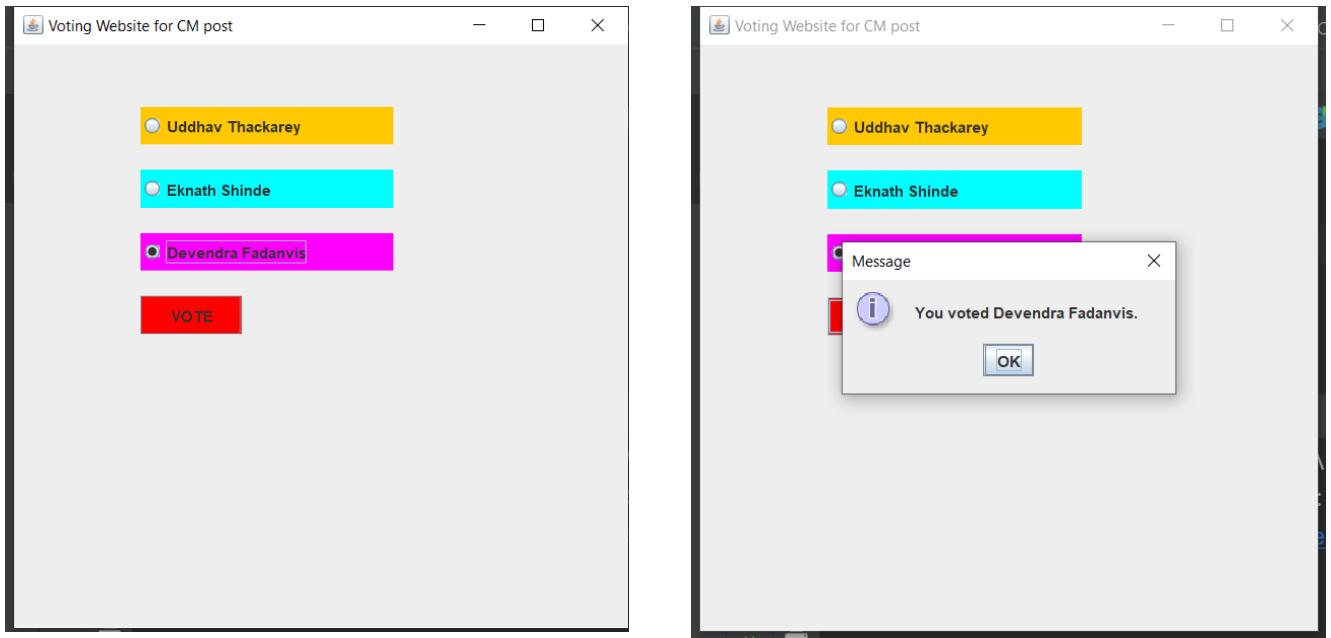
JButton b;
event3(){
    rb1 = new JRadioButton("Uddhav Thackarey");
    rb1.setBounds(100,50,200,30);
    rb1.setBackground(Color.orange);
    rb2 = new JRadioButton("Eknath Shinde");
    rb2.setBounds(100,100,200,30);
    rb2.setBackground(Color.CYAN);
    rb3 = new JRadioButton("Devendra Fadanvis");
    rb3.setBounds(100,150,200,30);
    rb3.setBackground(Color.MAGENTA);
    ButtonGroup bg = new ButtonGroup();
    JButton b4 = new JButton(new ImageIcon("C:\\Users\\manis\\Downloads\\new-1.jpg"));
    b4.setBackground(Color.orange);
    b4.setBounds(10,20,10,10);
    bg.add(rb1);bg.add(rb2);bg.add(rb3);
    b = new JButton("VOTE");
    b.setBounds(100,200,80,30);
    b.setBackground(Color.red);
    b.addActionListener(this);
    add(rb1);add(rb2);add(rb3);add(b);
    setTitle("Voting Website for CM post");
    setSize(500,500);
    setLayout(null);
    setVisible(true);
}
public void actionPerformed(ActionEvent e){
    if(rb1.isSelected()){
        JOptionPane.showMessageDialog(this,"You voted Uddhav Thackeray.");
    }
    if (rb2.isSelected()){
        JOptionPane.showMessageDialog(this,"You voted Eknath Shinde.");
    }
    if (rb3.isSelected()){
        JOptionPane.showMessageDialog(this,"You voted Devendra Fadanvis.");
    }
}

public static void main(String[] args) {
    new event3();
}
}

```

Output:-





Practical Performance (4)	Writeup & Oral (4)	Attendance (2)	Total (10)

Experiment no.7

Aim:- implementation of client-server Architecture using TCP/IP.

A)Client-program:-

Input:-

```
Package manishjadhav;

import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.net.InetAddress;
import java.net.Socket;
import java.net.UnknownHostException;

public class socketclient4 {
    private static Socket Socket;
    private static int i;
    public static void main(String[] args) throws
        UnknownHostException, IOException, ClassNotFoundException, InterruptedException{
        InetAddress host=InetAddress.getLocalHost();
        Socket socket=null;
        ObjectOutputStream oos=null;
        ObjectInputStream ois=null;
        for(int i=0;i<5;i++){
            {
                Socket=new Socket(host.getHostName(),14);
                oos=new ObjectOutputStream(Socket.getOutputStream());
                System.out.println("SENDING REQUEST SOCKET SERVER");
                oos.writeObject("Name: Manish Jadhav\n"+ "PRN:2030408246006\n" + "Roll no: 2201933");
                ois=new ObjectInputStream(Socket.getInputStream());
                String message=(String)
                    ois.readObject();
                System.out.println("Message:"+message);
                ois.close();
                oos.close();
                Thread.sleep(100);
            }
        }
    }
}
```

b)server-program:-**input:-**

```
package manishjadhav;

import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.net.ServerSocket;
import java.net.Socket;

public class socketserver4 {
    static ServerSocket server;
    static final int port=14;
    public static void main(String[]args)throws IOException,ClassNotFoundException{
        socketserver4 f=new socketserver4();
        server=new ServerSocket(port);
        while(true){
            System.out.println("WAITING FOR CLIENT REQUEST");
            Socket socket=server.accept();
            ObjectInputStream ois=new ObjectInputStream(socket.getInputStream());
            String message=(String)
                ois.readObject();
            System.out.println("Message received "+message);
            ObjectOutputStream oos= new ObjectOutputStream(socket.getOutputStream());
            oos.writeObject("Hi client,Name and PRN Received "+message);
            oos.close();
            socket.close();
            if(message.equalsIgnoreCase("exit"))
                break;
        }
        System.out.println("Shut down socket server!!!.....");
    }
}
```

Output:-

```
Run: socketserver4 × socketclient4 ×
C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" "-javaagent:C:\IBM\WAS\wlp\javaagent.jar" -Djava.util.logging.config.file=logging.properties
WAITING FOR CLIENT REQUEST
|
```

```
Run: socketserver4 × socketclient4 ×
▶ C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" "-javaagent:C:\IBM\WAS\wlp\javaagent.jar" -Djava.util.logging.config.file=logging.properties
SENDING REQUEST SOCKET SERVER
Message:Hi client,Name and PRN Received Name: Manish Jadhav
PRN:2030408246006
Roll no: 2201933
Process finished with exit code 0
```

Practical Performance (4)	Writeup & Oral (4)	Attendance (2)	Total (10)

Name:- Manish Shashikant Jadhav

Roll no.: - 2201933

PRN no.: - 2030408246006

Subject:- Advanced Java Programming Lab

Experiment no.8

Aim:- Implementation of Client-Server using UDP.

Program A) SERVER SIDE PRORGAM:-

Input:-

```
package com.AJPExperiments;
import java.net.*;
import java.util.Date;

public class sendtime {
    public static void main(String[] args) throws Exception{
        DatagramSocket b = new DatagramSocket();
        InetAddress address= InetAddress.getLocalHost();
        System.out.println("Server is ready to execute the client side program.");
        System.out.println("Please Continue");
        while (true){
            Thread.sleep(5000);
            Date currentDate = new Date();
            String s1 = currentDate.toString();
            byte arr[]=s1.getBytes();
            DatagramPacket dpack = new DatagramPacket(arr, arr.length,address,2000);
            b.send(dpack);
        }
    }
}
```

Output:-

The screenshot shows the IntelliJ IDEA interface with the terminal tab active. The command entered is:

```
"C:\Program Files\Java\jdk-17.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2\lib\idea_rt.jar=60005:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2\bin" -Dfile.encoding=UTF-8 -classpath c:\Users\manis\IdeaProjects\AJPCollege\out\production\AJPCollege com.AJPExperiments.sendtime
```

The output in the terminal is:

```
Server is ready to execute the client side program.  
Please Continue
```

At the bottom, it says "build completed successfully in 1 sec. 711 ms (a minute ago)".

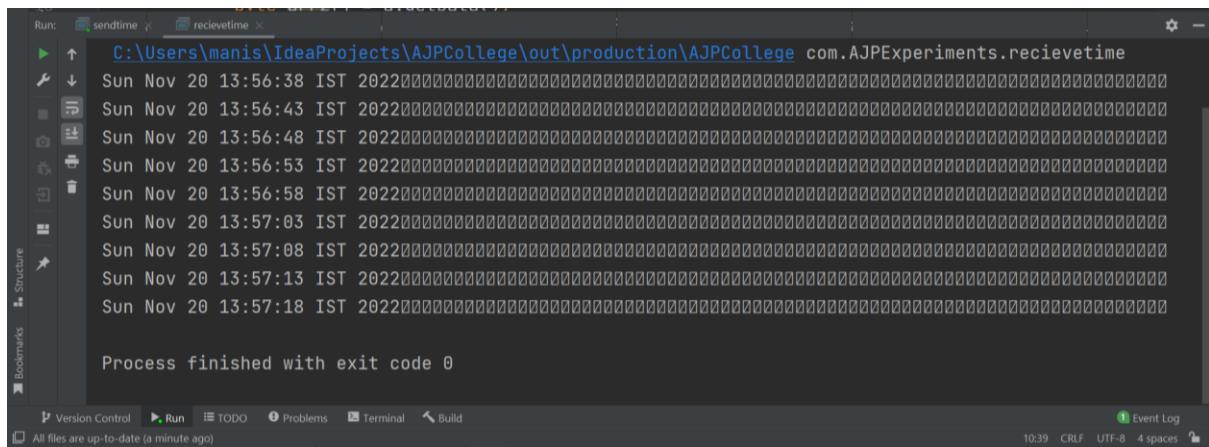
Program B) CLIENT SIDE PROGRAM:-

Input:-

```
package com.AJPExperiments;
import java.net.*;
import java.io.*;

public class recievetime {
    public static void main(String[] args) throws SocketException, IOException{
        DatagramSocket b = new DatagramSocket(2000);
        DatagramPacket a;
        for (int i = 0; i<9; i++){
            byte arr1[]=new byte[100];
            a = new DatagramPacket(arr1, arr1.length);
            b.receive(a);
            byte arr2[] = a.getData();
            String str = new String(arr2);
            System.out.println(str);
        }
    }
}
```

Output:-



```
Run: sendtime < recievetime
C:\Users\manis\IdeaProjects\AJPCollege\out\production\AJPCollege com.AJPExperiments.recievetime
Sun Nov 20 13:56:38 IST 2022
Sun Nov 20 13:56:43 IST 2022
Sun Nov 20 13:56:48 IST 2022
Sun Nov 20 13:56:53 IST 2022
Sun Nov 20 13:56:58 IST 2022
Sun Nov 20 13:57:03 IST 2022
Sun Nov 20 13:57:08 IST 2022
Sun Nov 20 13:57:13 IST 2022
Sun Nov 20 13:57:18 IST 2022

Process finished with exit code 0
```

Version Control Run TODO Problems Terminal Build Event Log
All files are up-to-date (a minute ago) 10:39 CRLF UTF-8 4 spaces

Practical Performance (4)	Writeup & Oral (4)	Attendance (2)	Total (10)

Name:- Manish Shashikant Jadhav

Roll no.: - 2201933

PRN no.: - 2030408246006

Subject:- Advanced Java Programming Lab

Experiment no.9

Aim:- Implementation of two-tier architecture using JDBC.

Program :-

Input:-

```
package com.AJPExperiments;
import javax.swing.JOptionPane;
import java.sql.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;

public class JDBCCTCreateTable {
    static String userid = "Manish", password="Leocor";
    static String url="jdbc.odbc bob";
    static Statement stmt;
    static Connection con;

    public static void main(String[] args) {
        JOptionPane.showMessageDialog(null,"JDBC Programming Creation of Voting
Table");
        int choice = -1;
        do{
            choice= getChoice();
            if (choice!=0){
                setSelected(choice);
            }
        }while (choice!=0);
        System.exit(0);
    }

    public static int getChoice(){
        String choice;
        int ch;
        choice= JOptionPane.showInputDialog(null,"1. Eknath Shinde\n"+
```

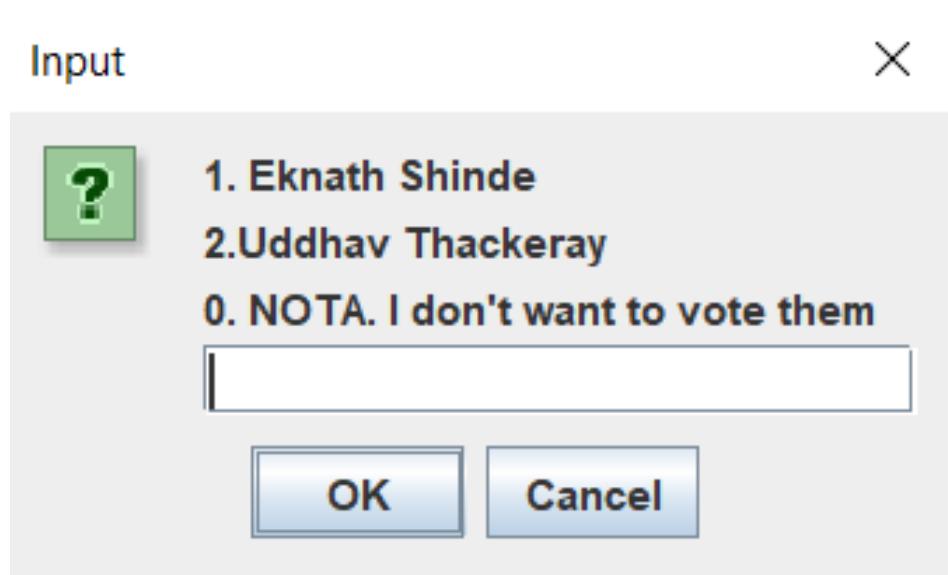
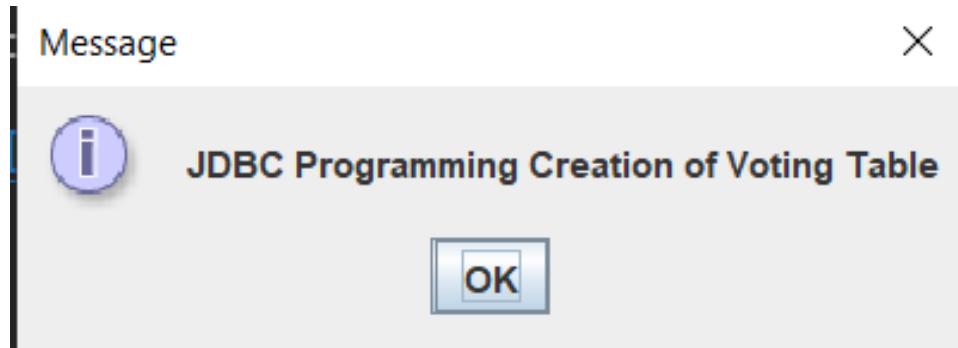
```

    "2.Uddhav Thackeray\n"+
    "0. NOTA. "+

    "I don't want to vote them");
ch=Integer.parseInt(choice);
return ch;
}
public static void getSelected(int choice){
if (choice==1){
    createVote();
}
if (choice==2){
    createVote2();
}
}
public static Connection getConnection() {
try {
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
} catch (java.lang.ClassNotFoundException e) {
    System.err.print("Class not Found");
    System.err.println(e.getMessage());
}
try {
    con = DriverManager.getConnection(url, userid, password);
} catch (SQLException ex) {
    System.out.println("SQLException: " + ex.getMessage());
}
return con;
}
public static void createVote(){
Connection con = getConnection();
String createString;
createString="create table vote(\"+\"ID INTEGER,\"+\"NAME VARCHAR(30))";
try{
    stmt=con.createStatement();
    stmt.executeUpdate(createString);
    stmt.close();
}catch (SQLException ex){
    System.err.println("SQL Exception: "+ex.getMessage());
}
 JOptionPane.showMessageDialog(null,"Voting Table Created");
}
public static void createVote2(){
    Connection con=getConnection();
    String createString;
    createString= "create table Vote2(\"+\"ID INTEGER,\"+\"NAME VARCHAR(20),\"+\"ID
    INTEGER)";
    try{

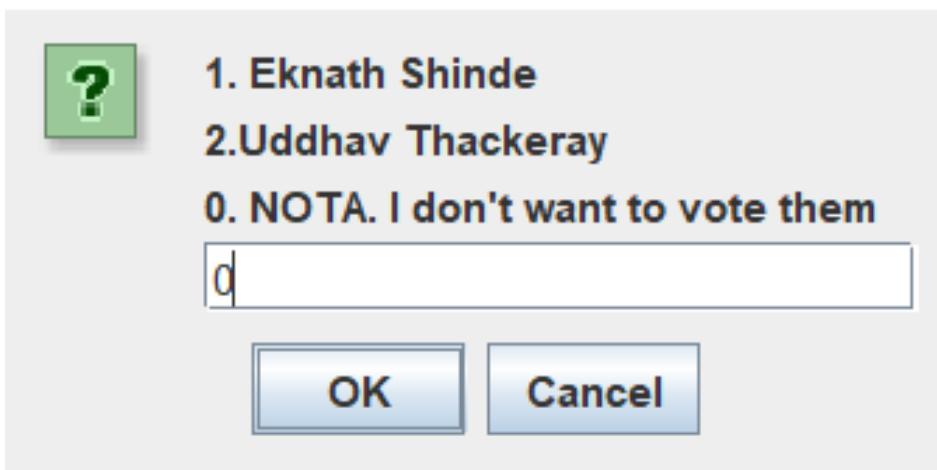
```

```
stmt = con.createStatement();
stmt.executeUpdate(createString);
stmt.close();
con.close();
}catch(SQLException ex){
    System.out.println("SQLException: "+ex.getMessage());
}
JOptionPane.showMessageDialog(null,"Voting Table 2 Created");
}
```

Output:-

Input

X



```
Run: JDBCCTCreateTable x
"C:\Program Files\Java\jdk-17.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2\lib\idea_rt.jar=60653:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2\bin" -Dfile.encoding=UTF-8 -classpath C:/Users/manis/IdeaProjects/AJPCollege/out/production/AJPCollege com.AJPExperiments.JDBCCTCreateTable

Process finished with exit code 0
```

Practical Performance (4)	Writeup & Oral (4)	Attendance (2)	Total (10)

Name:- Manish Shashikant Jadhav

Roll no.: - 2201933

PRN no.: - 2030408246006

Subject:- Advanced Java Programming Lab

Experiment no.10

Aim:- Implementation of Thread Life Cycle.

Program A):-

Input:-

```
package com.AJPExperiments;

class thread implements Runnable {
    public void run(){
        try{
            Thread.sleep(1500);
        }
        catch(InterruptedException e){
            e.printStackTrace();
        }
        System.out.println("State of thread1 while it called join() method on thread2 - "+
Test.thread1.getState());
        try{
            Thread.sleep(200);
        }catch(InterruptedException e){
            e.printStackTrace();
        }
    }
}

class Test implements Runnable{
    public static Thread thread1;
    public static Test obj;
    public static void main(String args[]){
        obj=new Test();
        thread1 = new Thread(obj);
        System.out.println("State of thread1 after creating it -"+ thread1.getState());
        thread1.start();
        System.out.println("State of thread1 after calling .start() method on it-
"+thread1.getState());
        thread1.getState();
    }
}
```

```

    }
    public void run(){
        thread myThread = new thread();
        Thread thread2 = new Thread(myThread);
        System.out.println("State of thread2 after creating it- "+thread2.getState());
        thread2.start();
        System.out.println("State of thread2 after calling .start() method on it-
"+thread2.getState());
        try{
            Thread.sleep(200);
        }catch(InterruptedException e){
            e.printStackTrace();
        }
        System.out.println("State of thread2 after calling .sleep() method on it-
"+thread2.getState());
        try{
            thread2.join();
        }catch (InterruptedException e){
            e.printStackTrace();
        }
        System.out.println("State of thread2 when it has finished it's execution-
"+thread2.getState());
    }
}

```

Output:-

```

Run: Test x
"C:\Program Files\Java\jdk-17.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2\lib\idea_rt.jar=53352:C:\Program Files\JetBrains\IntelliJ IDEA
Community Edition 2021.3.2\bin" -Dfile.encoding=UTF-8 -classpath C:\Users\manis\IdeaProjects\AJPCollege\out\production\AJPCollege com.AJPExperiments.Test
↑
State of thread1 after creating it -NEW
↓
State of thread1 after calling .start() method on it- RUNNABLE
State of thread2 after creating it -NEW
↓
State of thread2 after calling .start() method on it- RUNNABLE
↓
State of thread2 after calling .sleep() method on it- TIMED_WAITING
↓
State of thread1 while it called join() method on thread2 - WAITING
↓
State of thread2 when it has finished it's execution-TERMINATED

Process finished with exit code 0

```

Event Log

Version Control Run TODO Problems Terminal Build

All files are up-to-date (moments ago)

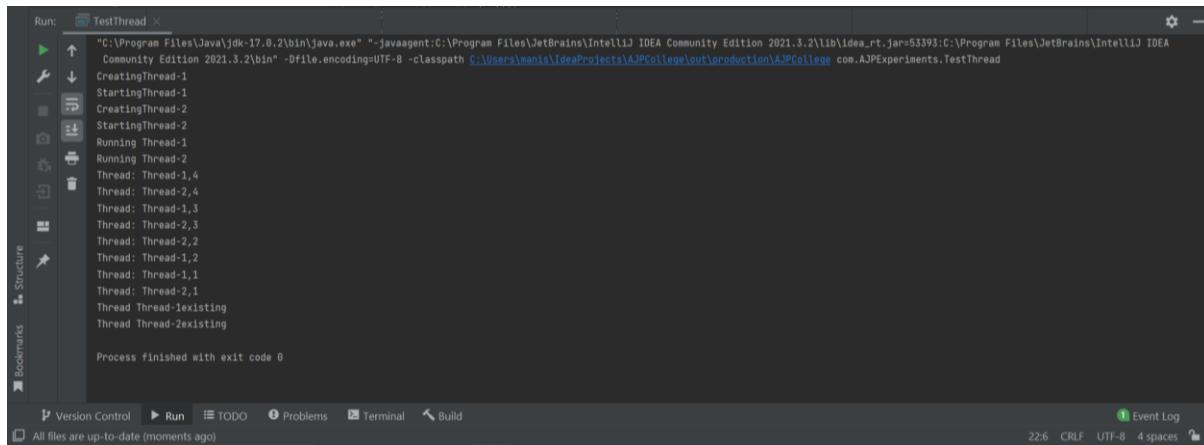
20:21 CRLF UTF-8 4 spaces

Program B):-**Input:-**

```
package com.AJPExperiments;

class RunnableDemo1 implements Runnable{
    private Thread t;
    private String threadName;
    RunnableDemo1(String name){
        threadName = name;
        System.out.println("Creating " + threadName);
    }
    public void run(){
        System.out.println("Running " + threadName);
        try{
            for(int i= 4; i>0; i--){
                System.out.println("Thread: " + threadName+"," +i);
                Thread.sleep(50);
            }
        }
        catch(InterruptedException e){
            System.out.println("Thread "+threadName+"interrupted.");
        }
        System.out.println("Thread "+threadName+"existing");
    }
    public void start(){
        System.out.println("Starting"+threadName);
        if(t==null){
            t=new Thread(this,threadName);
            t.start();
        }
    }
}
class TestThread{
    public static void main(String args[]){
        RunnableDemo1 R1= new RunnableDemo1("Thread-1");
        R1.start();
        RunnableDemo1 R2 = new RunnableDemo1("Thread-2");
        R2.start();
    }
}
```

Output:-



```
"C:\Program Files\Java\jdk-17.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2\lib\idea_rt.jar=53393:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2\bin" -Dfile.encoding=UTF-8 -classpath C:\Users\manis\IdeaProjects\AJPCollege\out\production\AJPCollege com.AJPEExperiments.TestThread
CreatingThread-1
StartingThread-1
CreatingThread-2
StartingThread-2
Running Thread-1
Running Thread-2
Thread: Thread-1,4
Thread: Thread-2,4
Thread: Thread-1,3
Thread: Thread-2,3
Thread: Thread-2,2
Thread: Thread-1,2
Thread: Thread-1,1
Thread: Thread-2,1
Thread Thread-lexisting
Thread Thread-2existing

Process finished with exit code 0
```

Practical Performance (4)	Writeup & Oral (4)	Attendance (2)	Total (10)

Experiment no.11

Aim:- Implementation of Runnable Interface.

Program A):-

Input:-

```
package com.AJPExperiments;

class MyRunnableThread implements Runnable {
    public static int myCount =8;
    public MyRunnableThread(){
    }
    public void run(){
        while(MyRunnableThread.myCount>=0){
            try{
                System.out.println("Expl Thread: "+(--MyRunnableThread.myCount));
                Thread.sleep(100);
            }catch (InterruptedException iex){
                System.out.println("Exception in thread: "+iex.getMessage());
            }
        }
    }
    class RunMyThread{
        public static void main(String args[]){
            System.out.println("Starting Main Thread...");
            MyRunnableThread mrt = new MyRunnableThread();
            Thread t = new Thread(mrt);
            t.start();
            while(MyRunnableThread.myCount>=0){
                try{
                    System.out.println("Main Thread: "+(--MyRunnableThread.myCount));
                    Thread.sleep(100);
                }catch(InterruptedException iex){
                    System.out.println("Exception in main thread: "+iex.getMessage());
                }
            }
            System.out.println("End of Main Thread... ");
        }
    }
}
```

Output:-

The screenshot shows a run window from an IDE. The title bar says "Run: RunMyThread". The main area displays the following text:

```
.encoding=UTF-8 -classpath C:\Users\manis\IdeaProjects\A.
Starting Main Thread...
Main Thread: 7
Expl Thread: 6
Main Thread: 5
Expl Thread: 4
Main Thread: 3
Expl Thread: 2
Main Thread: 1
Expl Thread: 0
Main Thread: -1
End of Main Thread...

Process finished with exit code 0
```

The bottom status bar shows "Build completed successfully in 1 sec, 628 ms (2 minutes ago)".

Program B):-

Input:-

```
package com.AJPExperiments;
```

```
class MultiThread implements Runnable{
    public void run(){
        for(int i=0;i<9;i++){
            System.out.println("Thread " + Thread.currentThread().getId()+" is running");
        }
    }
}
class RunThread{
    public static void main(String args[]){
        Thread object = new Thread(new MultiThread());
        object.start();
        for(int i = 0; i<9;i++){
            System.out.println("Main Thread id:"+Thread.currentThread().getId());
        }
    }
}
```

Output:-

Practical Performance (4)	Writeup & Oral (4)	Attendance (2)	Total (10)