CENTRAL CALCUTTA POLYTECHNIC

21, Convent Road, Philips, Sealdah, Kolkata, West Bengal 700014

DEPT. : COMPUTER SCIENCE AND TECHNOLOGY

- NAME : SUMAN MONDAL
- ROLL : DCCPCSTS6
- NUMBER : 10005537
- REG NUMBER : D192005242
- SUBJECT : ADVANCED JAVA PROGRAMMING
- SESSION : 2021 - 2022
- EMAIL : SUMAN.MONDAL@OUTLOOK.IN

Contents

Chapter 1

AWT

1.1 Frame with Title and Label

Source Code :

```
import java.awt.*;

public class Q01 {
    public static void main(String args[]) {
        Frame f = new Frame("Frame Title");
        f.setSize(300, 200);
        Label l = new Label("text is here");
        f.add(l);
        f.setVisible(true);
    }
}
```

Program Output :

1.2 Adding Button and Flow Layout

Source Code :

```java
import java.awt.*;

public class Q02 {
    public static void main(String args[]) {
        Frame f = new Frame("Hello World");
        f.setSize(300, 200);
        f.setLayout(new FlowLayout());
        Button b = new Button("YES");
        Button m = new Button("NO");
        f.add(b);
        f.add(m);
        f.setVisible(true);
    }
}
```

Program Output :

1.3 Adding Grid Layout

Source Code :

```java
import java.awt.*;

public class Q03 {
    public static void main(String args[]) {
        Frame f = new Frame("hello world");
        f.setSize(400, 200);
        f.setLayout(new GridLayout(1, 2));
        f.add(new Button("YES"));
        f.add(new Button("NO"));
        f.setVisible(true);
```

```
    }
}
```

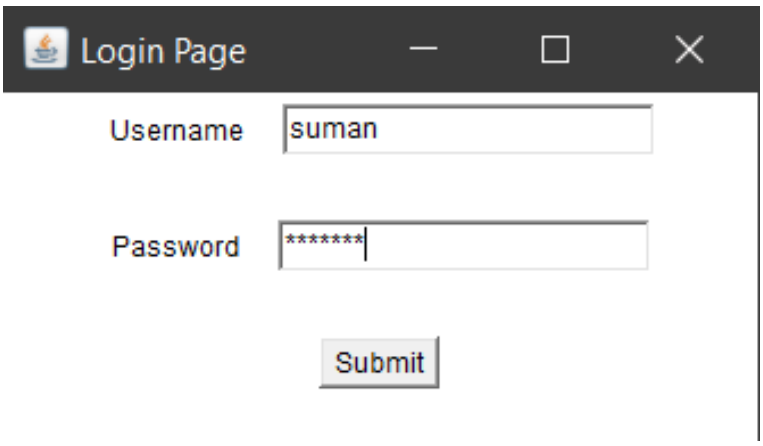
Program Output :

1.4 Create Login Form

Source Code :

```
import java.awt.*;

class Q04 {
    public static void main(String[] args) {
        Frame ob1 = new Frame("Login Page");
        ob1.setSize(350, 200);
        ob1.setLayout(new GridLayout(3, 0));
        Panel idPanel = new Panel();
        ob1.add(idPanel);
        idPanel.setLayout(new FlowLayout());
        idPanel.add(new Label("Username"));
        idPanel.add(new TextField(20));
        Panel passPanel = new Panel();
        ob1.add(passPanel);
        passPanel.setLayout(new FlowLayout());
        passPanel.add(new Label("Password"));
        TextField password = new TextField(20);
        password.setEchoChar('*');
        passPanel.add(password);
        Panel submitPanel = new Panel();
        ob1.add(submitPanel);
        submitPanel.setLayout(new FlowLayout());
        submitPanel.add(new Button("Submit"));
        ob1.setVisible(true);
    }
}
```

Program Output :

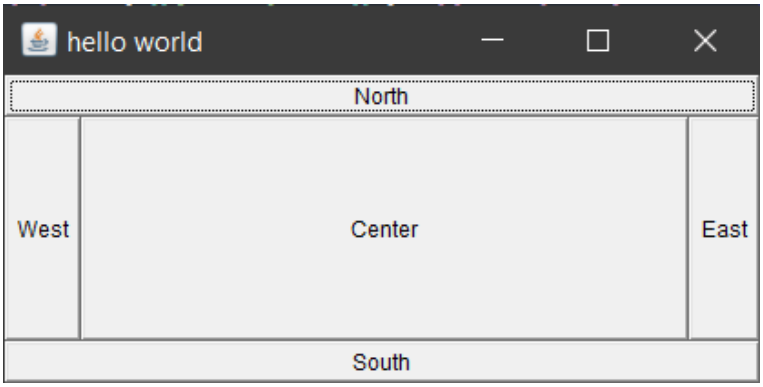


1.5 Adding Border Layout

Source Code :

```
import java.awt.*;

public class Q05 {
    public static void main(String args[]) {
        Frame f = new Frame("hello world");
        f.setSize(300, 150);
        f.setLayout(new BorderLayout());
        f.add(new Button("North"), BorderLayout.NORTH);
        f.add(new Button("South"), BorderLayout.SOUTH);
        f.add(new Button("East"), BorderLayout.EAST);
        f.add(new Button("West"), BorderLayout.WEST);
        f.add(new Button("Center"), BorderLayout.CENTER);
        f.setVisible(true);
    }
}
```

Program Output :



1.6 Calculator Using Panel

Source Code :

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

public class Q06 {
    public static void main(String[] args) {
        JFrame ob1 = new JFrame("Calculator");
        ob1.setSize(250, 300);
        Panel buttonPanel = new Panel();
        buttonPanel.setLayout(new GridLayout(5, 3, 5, 5));
        for (int i = 1; i < 10; i++) {
            buttonPanel.add(new Button(i + ""));
        }
        String[] extra = { "*", "/", "0", "+", "-", "=" };
        for (int i = 0; i < extra.length; i++) {
            buttonPanel.add(new Button(extra[i]));
        }
        ob1.add(new TextArea(2, 20), BorderLayout.NORTH);
        ob1.add(buttonPanel, BorderLayout.CENTER);
        ob1.setVisible(true);
    }
}
```

Program Output :

1.7 Adding Checkbox

Source Code :

```
import java.awt.*;

public class Q07 {
    public static void main(String args[]) {
        Frame f = new Frame("BRANCH");
        f.setSize(300, 100);
        f.setLayout(new GridLayout(6, 2));
        f.add(new Checkbox("CST", null, true));
        f.add(new Checkbox("ETC"));
        f.add(new Checkbox("ME"));
        f.setVisible(true);
    }
}
```

Program Output :

1.8 Copy a Textfield Content to Another Textfield Using Event Handling

Source Code :

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

public class Q08 extends Frame {
    TextField t1 = new TextField(), t2 = new TextField();

    Q08() {
        setSize(300, 150);
        setLayout(new FlowLayout());
        Button b = new Button("Copy- ->");
        add(t1);
        add(b);
        add(t2);
        b.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                t2.setText(t1.getText());
            }
        });
        setVisible(true);
    }

    public static void main(String args[]) {
        new Q08();
    }
}
```

Program Output :

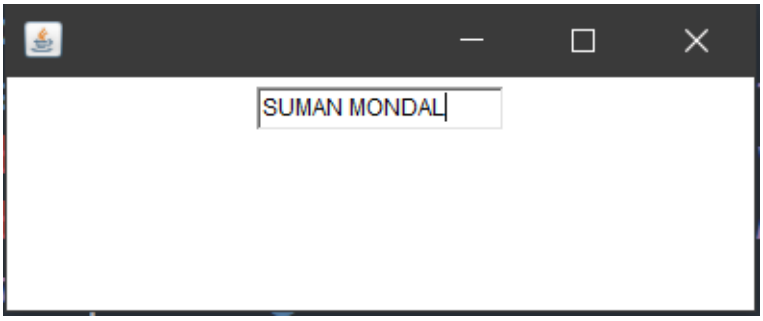
1.9 Textfield Content Always in Upper Case

Source Code :

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

public class Q09 extends Frame {
    TextField t = new TextField(15);

    Q09() {
        setSize(200, 150);
        setLayout(new FlowLayout());
        add(t);
        t.addKeyListener(new KeyListener() {
            public void keyPressed(KeyEvent e) {
            }

            public void keyReleased(KeyEvent e) {
                int cp = t.getCaretPosition();
                t.setText(t.getText().toUpperCase());
                t.setCaretPosition(cp);
            }

            public void keyTyped(KeyEvent e) {
            }
        });
        setVisible(true);
    }

    public static void main(String args[]) {
        new Q09();
    }
```

```
}
```

Program Output :



1.10 User Details Entry using AWT Textfield, Checkbox Group, Choice Object and Event Handling

Source Code :

```
import java.awt.*;
import java.awt.event.*;
public class Demoevent implements WindowListener
{
    public Demoevent()
    {
        GridLayout grid = new GridLayout(1, 3, 5, 5);

        Frame f = new Frame("Sign Up");
        f.setSize(600, 400);
        f.setLayout(new FlowLayout());
        f.setVisible(true);

        Panel input = new Panel();
        input.setLayout(new GridLayout(3, 1, 5, 5));

        Panel ipName = new Panel();
        ipName.setLayout(grid);
        ipName.add(new Label("Name"));
        TextField fName = new TextField(15);
        TextField lName = new TextField(15);
        ipName.add(fName);
        ipName.add(lName);

        Panel ipFather = new Panel();
        ipFather.setLayout(grid);
        ipFather.add(new Label("Father's Name"));
```

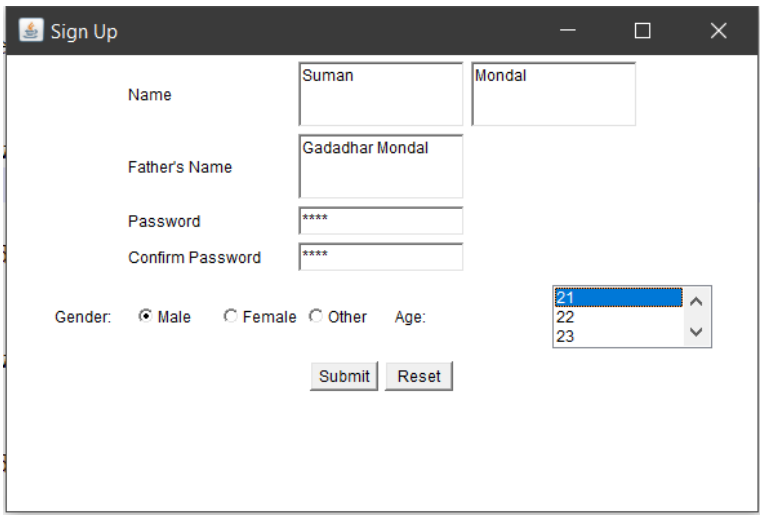
```
TextField fname = new TextField(15);
ipFather.add(fname);
ipFather.add(new Label());

Panel paswrd = new Panel();
paswrd.setLayout(new GridLayout(2,3,5,5));

paswrd.add(new Label("Password"));
TextField pass1 = new TextField(15);
pass1.setEchoChar('*');
paswrd.add(pass1);
paswrd.add(new Label());

paswrd.add(new Label("Confirm Password"));
TextField pass2 = new TextField(15);
pass2.setEchoChar('*');
paswrd.add(pass2);
paswrd.add(new Label());

input.add(ipName);
input.add(ipFather);
input.add(paswrd);

Panel chinput = new Panel();
chinput.setLayout(new FlowLayout(FlowLayout.LEFT, 5, 5));

Panel gender = new Panel();
gender.setLayout(new GridLayout(1, 4, 5, 5));
gender.add(new Label("Gender:"));
CheckboxGroup gendergrp = new CheckboxGroup();
gender.add(new Checkbox("Male", gendergrp, true));
gender.add(new Checkbox("Female", gendergrp, false));
gender.add(new Checkbox("Other", gendergrp, false));

Panel age = new Panel();
age.setLayout(new GridLayout(1, 2));
List ageList = new List(3, false);
for(int i =10; i<30; i++)
    ageList.add(""+i);
age.add(new Label("Age:"));
age.add(ageList);
```

```
chinput.add(gender);
chinput.add(age);

Panel btnIp = new Panel();
btnIp.setLayout(new GridLayout(1, 2, 5, 5));
Button smt = new Button("Submit");
smt.addActionListener(new ActionListener(){
    @Override
    public void actionPerformed(ActionEvent e){
        Object ob = e.getSource();
        if(ob == smt){
            Frame f1 = new Frame();
            f1.setLayout(new GridLayout(4, 2));
            f1.setSize(400, 150);
            f1.setVisible(true);

            if(pass1.getText().equals(pass2.getText())){
                f1.setTitle("New Entry");
                f1.add(new Label("USER DETAILS:"));
                String user = new String(fName.getText()+"
                ↪  "+lName.getText()+" ,
                ↪  "+gendergrp.getSelectedCheckbox().getLabel()+"
                ↪  "+ageList.getSelectedItem());
                String password = null;
                for(int i=0;i<pass1.getText().length();i++){
                    if(i==0)
                        password =
                        ↪  pass1.getText().substring(0, 1);
                    else if(i==pass1.getText().length()-1)
                        password =
                        ↪  password+""+pass1.getText().substring(i
                        ↪  i+1);
                    else
                        password = password+"*";
                }
                f1.add(new Label(user));
                f1.add(new Label(password));
            }
            else{
                f1.setTitle("Error");
                f1.add(new Label("Conformation Password not
                ↪  matching!!"));
```

```
                }
                f1.addWindowListener(new WindowAdapter(){
                    @Override
                    public void windowClosing(WindowEvent e){
                        f1.dispose();
                    }
                });
            }
        }
    });
    btnIp.add(smt);

    Button rst = new Button("Reset");
    rst.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            Object ob = e.getSource();
            if(ob==rst){
                fName.setText("");
                lName.setText("");
                fname.setText("");
                pass1.setText("");
                pass2.setText("");
            }
        }
    });
    btnIp.add(rst);

    f.add(input, BorderLayout.NORTH);
    f.add(chinput, BorderLayout.CENTER);
    f.add(btnIp, BorderLayout.SOUTH);

    f.addWindowListener(new WindowAdapter() {
        @Override
        public void windowClosing(WindowEvent e){
            f.dispose();
        }
    });
}

@Override
public void windowOpened(WindowEvent e){}
```

```
    @Override
    public void windowClosing(WindowEvent e){}

    @Override
    public void windowClosed(WindowEvent e){}

    @Override
    public void windowIconified(WindowEvent e){}

    @Override
    public void windowDeiconified(WindowEvent e){}

    @Override
    public void windowActivated(WindowEvent e){}

    @Override
    public void windowDeactivated(WindowEvent e){}

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

Program Output :



Chapter 2

Socket Programming

2.1 TCP Socket Program

2.1.1 Server Side

Source Code :

```
import java.io.*;
import java.net.*;

public class Q01ServerSide {
    public static void main(String[] args) throws Exception {
        final int port = 7999;
        ServerSocket pServerS = new ServerSocket(port);
        System.out.println("Waiting for Connection @localhost:" + port);
        Socket servSock = pServerS.accept();
        System.out.println("Connection Established");
        BufferedReader clientInput = new BufferedReader(new
        ↪  InputStreamReader(servSock.getInputStream()));
        PrintWriter clientOutput = new PrintWriter(servSock.getOutputStream(),
        ↪  true);
        int data = Integer.parseInt(clientInput.readLine());
        System.out.println("Client Input:" + data);
        int fact = 1;
        for (int i = 2; i <= data; i++) {
            fact *= i;
        }
        clientOutput.println(fact);
        System.out.println("Server Response sent to client: " + fact);
    }
}
```

Program Output :

```
E:\Assignment\codes\socket>java Q01ServerSide
Waiting for Connection @localhost:7999
Connection Established
Client Input:3
Server Response sent to client: 6
```

2.1.2 Client Side

Source Code :

```
import java.io.*;
import java.net.*;
import java.awt.*;
import java.awt.event.*;

public class Q01ClientSide {
    public static void main(String[] args) throws Exception {
        final int port = 7999;
        Socket cliSock = new Socket("localhost", port);
        System.out.println("Connected to SERVER @localhost:" + port);
        PrintWriter servInput = new PrintWriter(cliSock.getOutputStream(), true);
        BufferedReader servResp = new BufferedReader(new
        ↪  InputStreamReader(cliSock.getInputStream()));
        BufferedReader userInput = new BufferedReader(new
        ↪  InputStreamReader(System.in));
        System.out.println("Enter and integer: ");
        String data = userInput.readLine();
        servInput.println(data);
        System.out.println("Sent to server: " + data);
        String fact = servResp.readLine();
        System.out.println("Recieved from server: " + fact);
        cliSock.close();
    }
}
```

Program Output :

```
E:\Assignment\codes\socket>java Q01ClientSide
Connected to SERVER @localhost:7999
Enter and integer:
3
Sent to server: 3
Recieved from server: 6
```

2.2 TCP Socket Program for Multiple Client using Threads

2.2.1 Server Side

Source Code :

```
import java.io.*;
import java.net.*;

class Q02ServerSide {
    public static void main(String argv[]) throws Exception {
        ServerSocket welcomeSocket = new ServerSocket(6789);
        System.out.println("Server Ready");
        while (true) {
            Socket serverEnd = welcomeSocket.accept();
            System.out.println("Request accepted");
```

```
            new Handler(serverEnd);
        }
    }
}

class Handler implements Runnable {
    Socket serverEnd;

    Handler(Socket s) {
        this.serverEnd = s;
        new Thread(this).start();
        System.out.println("A Thread created");
    }

    public void run() {
        try {
            BufferedReader fromClient = new BufferedReader(new
            ↪  InputStreamReader(serverEnd.getInputStream()));
            PrintWriter toClient = new PrintWriter(serverEnd.getOutputStream(),
            ↪  true);
            while (true) {
                int n = Integer.parseInt(fromClient.readLine());
                System.out.println("Received " + n);
                if (n == -1) {
                    serverEnd.close();
                    break;
                }
                int fact = 1;
                for (int i = 2; i <= n; i++) {
                    fact *= i;
                }
                toClient.println(fact);
                System.out.println("Sent: " + fact);
            }
        } catch (IOException e) {
        }
    }
}
```

Program Output :

```
E:\Assignment\codes\socket>java Q02ServerSide
Server Ready
Request accepted
A Thread created
Received 3
Sent: 6
Received 4
Sent: 24
Received 5
Sent: 120
Received 6
Sent: 720
Received 7
```

2.2.2 Client Side

Source Code :

```
import java.io.*;
import java.net.*;

public class Q02ClientSide {
    public static void main(String argv[]) throws Exception {
        String fact;
        Socket clientEnd = new Socket("localhost", 6789);
        System.out.println("connected to localhost at port 6789");
        PrintWriter toServer = new PrintWriter(clientEnd.getOutputStream(), true);
        BufferedReader fromServer = new BufferedReader(new
        ↪  InputStreamReader(clientEnd.getInputStream()));
        BufferedReader fromUser = new BufferedReader(new
        ↪  InputStreamReader(System.in));
        while (true) {
            System.out.print("Enter an integer: ");
            String n = fromUser.readLine();
            toServer.println(n);
            System.out.println("Sent to server: " + n);
            if (n.equals("-1"))
                break;
            fact = fromServer.readLine();
            System.out.println("Received from server: " + fact);
        }
        clientEnd.close();
    }
}
```

Program Output :

```
E:\Assignment\codes\socket>java Q02ClientSide
connected to localhost at port 6789
Enter an integer: 3
Sent to server: 3
Received from server: 6
Enter an integer: 4
Sent to server: 4
Received from server: 24
Enter an integer: 5
Sent to server: 5
Received from server: 120
Enter an integer: 6
Sent to server: 6
Received from server: 720
Enter an integer: 7
Sent to server: 7
```

2.3 UDP Socket Program

2.3.1 Server Side

Source Code :

```
import java.net.*;
import java.io.*;

public class Q03ServerSide {
    public static void main(String args[]) throws Exception {
        DatagramSocket socket = new DatagramSocket(5000);
        System.out.println("Server Ready");
        while (true) {
            byte[] rbuf = new byte[10];
            DatagramPacket rpkt = new DatagramPacket(rbuf, rbuf.length);
            socket.receive(rpkt);
            System.out.println("Received a packet");
            new Handler(rpkt, socket);
        }
    }
}

class Handler implements Runnable {
    DatagramSocket socket;
    DatagramPacket pkt;

    Handler(DatagramPacket pkt, DatagramSocket socket) {
        this.pkt = pkt;
        this.socket = socket;
        new Thread(this).start();
        System.out.println("A thread created");
    }

    public void run() {
        try {
```

```
            byte[] sbuf = new byte[10];
            String data = new String(pkt.getData(), 0, pkt.getLength());
            InetAddress addr = pkt.getAddress();
            int port = pkt.getPort();
            int fact = 1, n = Integer.parseInt(data);
            System.out.println("Received: " + n + " from " + addr + ":" + port);
            for (int i = 2; i <= n; i++)
                fact *= i;
            sbuf = String.valueOf(fact).getBytes();
            DatagramPacket spkt = new DatagramPacket(sbuf, sbuf.length, addr,
                    port);
            socket.send(spkt);
            System.out.println("Sent: " + fact);
        } catch (IOException e) {
        }
    }
}
```

Program Output :

```
E:\Assignment\codes\socket>javac Q03ServerSide.java

E:\Assignment\codes\socket>java Q03ServerSide
Server Ready
Received a packet
A thread created
Received: 3 from /127.0.0.1:50754
Sent: 6
Received a packet
A thread created
Received: 2 from /127.0.0.1:50754
Sent: 2
```

2.3.2 Client Side

Source Code :

```
import java.net.*;
import java.io.*;

public class Q03ClientSide {
    public static void main(String args[]) throws Exception {
        byte[] rbuf = new byte[1024], sbuf = new byte[1024];
        BufferedReader fromUser = new BufferedReader(new
         ↪  InputStreamReader(System.in));
        DatagramSocket socket = new DatagramSocket();
        InetAddress addr = InetAddress.getByName(args[0]);
        while (true) {
            System.out.print("Enter an Integer: ");
            String data = fromUser.readLine();
            sbuf = data.getBytes();
            DatagramPacket spkt = new DatagramPacket(sbuf, sbuf.length, addr,
                    5000);
            socket.send(spkt);
            System.out.println("Sent to server: " + data);
```

```
            if (data.equals("-1"))
                break;
            DatagramPacket rpkt = new DatagramPacket(rbuf, rbuf.length);
            socket.receive(rpkt);
            data = new String(rpkt.getData(), 0, rpkt.getLength());
            System.out.println("Received from server: " + data);
        }
        socket.close();
    }
}
```

Program Output :

```
E:\Assignment\codes\socket>java Q03ClientSide localhost
Enter an Integer: 3
Sent to server: 3
Received from server: 6
Enter an Integer: 2
Sent to server: 2
Received from server: 2
Enter an Integer:
```

Chapter 3

Swing

3.1 JPasswordField

Source Code :

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

public class Q01 {
    public static void main(String args[]) {
        final JFrame f = new JFrame("JPasswordField ");
        f.setSize(200, 150);
        Container c = f.getContentPane();
        c.setLayout(new FlowLayout());
        c.add(new JLabel("Password"));
        final JPasswordField pf = new JPasswordField(10);
        c.add(pf);
        f.setVisible(true);
    }
}
```

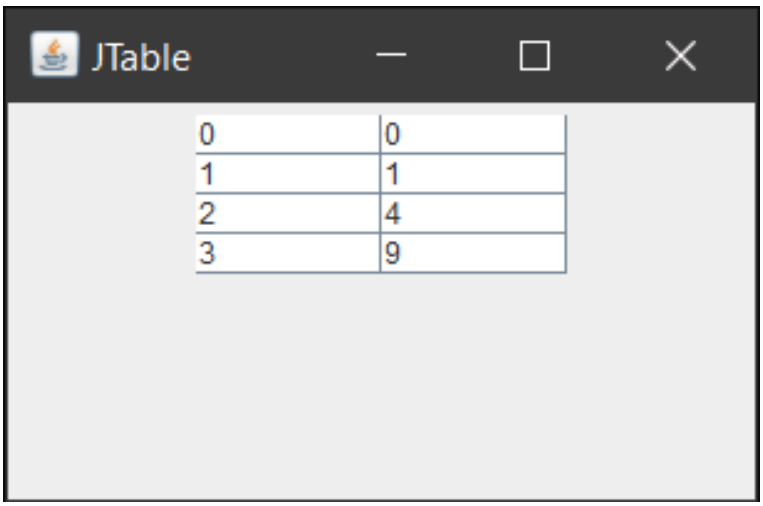
Program Output :

3.2 JTable

Source Code :

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

public class Q02 {
    public static void main(String args[]) {
        final JFrame f = new JFrame("JTable ");
        f.setSize(250, 150);
        Container c = f.getContentPane();
        c.setLayout(new FlowLayout());
        int row = 4, col = 2;
        JTable table = new JTable(row, col);
        for (int i = 0; i < row; i++) {
            table.setValueAt(i, i, 0);
            table.setValueAt(i * i, i, 1);
        }
        c.add(table);
        f.setVisible(true);
    }

}
```

Program Output :



3.3 JPasswordField

Source Code :

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

public class Q03 {
    public static void main(String args[]) throws Exception {
```

```
        final JFrame f = new JFrame("JProgressBar ");
        f.setSize(200, 80);
        Container c = f.getContentPane();
        c.setLayout(new FlowLayout());
        JProgressBar pb = new JProgressBar(0, 100);
        pb.setStringPainted(true);
        c.add(pb);
        for (int i = 0; i <= 100; i++) {
            pb.setValue(i);
            Thread.sleep(100);
        }
        f.setVisible(true);
    }
}
```

Program Output :