

## Java Source Code:

```
package omkaranuk
```

```
import java.awt.FlowLayout;
```

```
public class Image extends JFrame {
```

```
private JLabel label1;
```

```
private JLabel label2;
```

```
Image() throws IOException {
```

```
setLayout(new FlowLayout());
```

```
File file = new
```

```
File("D:\\13-2\\network with Java\\image
```

```
\\IEEE-logo.jpg");
```

```
BufferedImage image1 = ImageIO.read(file);
```

```
ImageIcon imageIcon = new ImageIcon(image1);
```

```
label1 = new JLabel(imageIcon);
```

```
add(label1)
```

```
File file2 = new File("D:\\13-2\\network with
```

```
Java\\Image\\post-logo.png");
```

```
BufferedImage image2 = ImageIO.read(file);
```

```
ImageIcon imageIcon2 = new ImageIcon(image2);
```

```
image2);
```

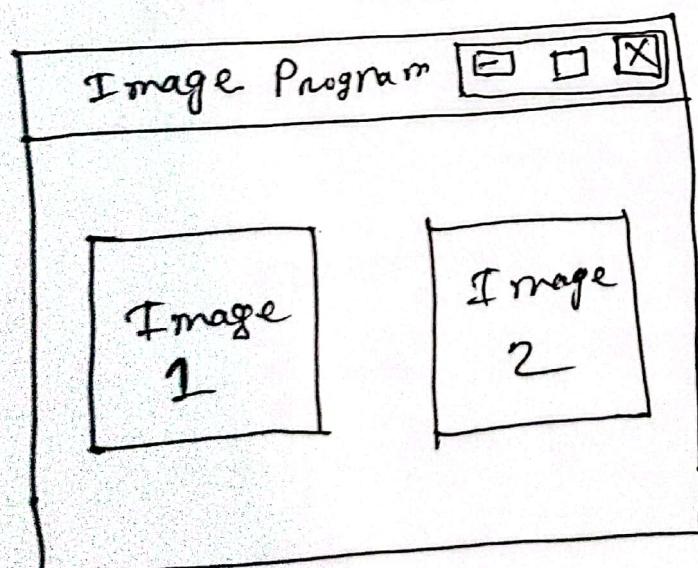
```
label2 = new JLabel(imageIcon2);
```

```
add(label2);
```

```
}
```

```
public static void main (String args[]) throws  
    IOException {  
    Image gui = new Image();  
    gui.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    gui.setVisible(true);  
    gui.setSize(500,500);  
    gui.setTitle("Image Program");  
}
```

## Output

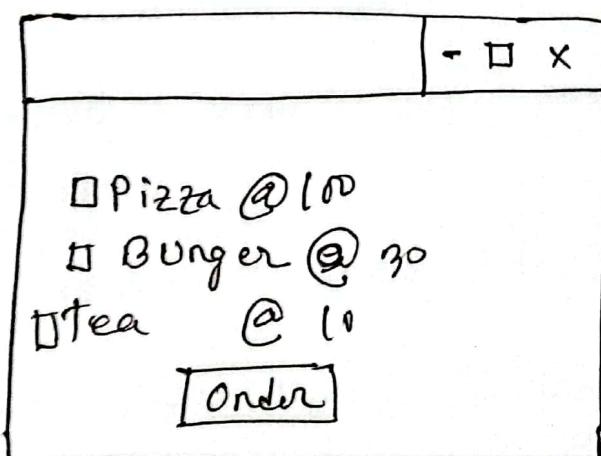


### Source Code:

```
Package omanfawz;  
import javax.swing;  
import java.awt.swing;  
public class BillGeneration extends JFrame  
    implements ActionListener {  
  
    JLabel l;  
    JCheckBox cb1, cb2, cb3;  
    JButton b;  
  
    BillGeneration () {  
        l = new JLabel ("Food ordering system");  
        l.setBounds (50, 50, 300, 20);  
        cb1 = new JCheckBox ("Pizza @ 100");  
        cb2 = new JCheckBox ("Burger @ 30");  
        cb3 = new JCheckBox ("Tea @ 10");  
  
        b = new JButton ("Order");  
        b.setBounds (100, 250, 80, 30);  
        b.addActionListener (this);  
        add (b);  
        setVisible (true);  
    }  
  
    public void actionPerformed (ActionEvent e) {  
        float amount = 0;  
        String msg = "Selected items : ";  
        if (cb1.isSelected ()) {  
            amount += 100;  
            msg += "Pizza @ 100";  
        }  
        if (cb2.isSelected ()) {  
            amount += 30;  
            msg += "Burger @ 30";  
        }  
        if (cb3.isSelected ()) {  
            amount += 10;  
            msg += "Tea @ 10";  
        }  
        JOptionPane.showMessageDialog (null, "Total Amount : " + amount + "\n" + msg);  
    }  
}
```

```
if (cb1.isSelected ()) {  
    amount += 10;  
    msg += " Pizza: 10 @ \n";  
}  
if (cb3.isSelected ()) {  
    amount += 10;  
    msg += " Tea : 10 @ \n";  
}  
msg += "----- \n";  
msg += "Total : " + amount;  
JOptionPane.showMessageDialog (this, msg);  
}  
public static void main (String [] args) {  
    new BillGeneration ();  
}
```

Output:



## SOURCE CODE:

Package Main.Panel;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.BorderLayout;

import java.awt.Container;

import java.awt.Label;

import java.awt.Panel;

import java.awt.TextField;

public class Form implements ActionListener {

private static Label success;

private static JFrame frame;

private static JButton button;

private static JTextField userText;

public static void main(String args[]) {

frame = new JFrame();

panel = new JPanel();

frame.add(panel);

panel.setLayout(null);

label1 = new JLabel("Name");

label1.setBounds(10, 10, 80, 25);

label2 = new JLabel("Roll");

label2.setBounds(10, 60, 80, 25);

panel.add(label2);

```

UserText1 = new JTextField ("Enter your name");
userText1.setBounds (100, 10, 200, 25);
panel.add (userText1);

UserText2 = new JTextField ("Enter your
roll");
userText2.setBounds (100, 110, 80, 25);
panel.add (userText2);

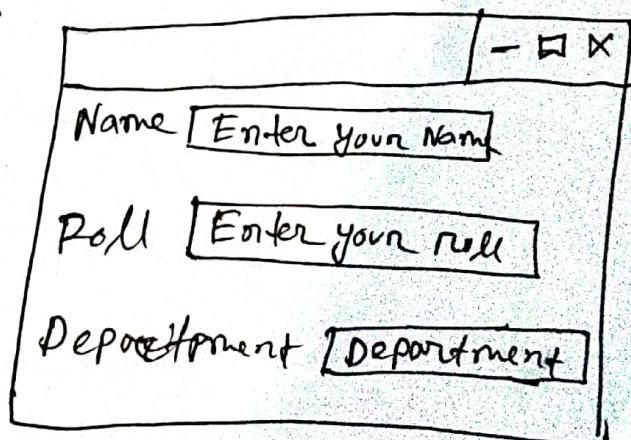
UserText3 = new JTextField ("Enter your
department");
panel.add (userText3);

}

@Override
public void actionPerformed (ActionEvent e) {
    success.setText ("Save. Successfully");
}

```

Output:



## Java Source Code:

```
Package omar;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.ActionListener;
import javax.swing.*;

public class calculator extends JFrame implements
    ActionListener {
    JButton b10, b11, b12, b13, b14, b15,
    JButton b[] = new JButton[10];
    int i, n, n1, n2;
    JTextField res;
    char op;
    public calculator() {
        super("calculator");
        setLayout(new BorderLayout());
        for (int i=0; i<=9; i++) {
            b[i] = new JButton(i + "");
            p.add(b[i]);
            b[i].addActionListener(this);
        }
        b10 = new JButton("+");
        b10.addActionListener(this);
        b10.add(res);
    }
}
```

```
b14 = new JButton("B");
p.add(b14);
b14.addActionListener(this);
b15 = new JButton("C");
p.add(B15);
b15.addActionListener(this);
}
public void actionPerformed(ActionEvent ae){
    JButton pb=(JButton)ae.getSource();
    if(pb==B15){
        n1=n2=' ';
        res.setText(" ");
    }else if(pb==b14){
        n2=Integer.parseInt(res.getText());
        eval();
        res.setText(" " +n);
    }else{
        boolean opf=false;
        if(pb==b10){
            op='+'; opf=true;
        }
        if(pb==b12)
            op='*';
            opf=true;
    }
}
```

```
else {
    n1 = Integer.parseInt(res.getText());
    res.setText(" ");
}

int eval() {
    switch(OP) {
        case '+':
            r = n1 + n2;
            break;
        case '-':
            r = n1 - n2;
            break;
        case '*':
            r = n1 * n2; break;
        case '/':
            r = n1 / n2;
            break;
    }
    return r;
}

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

Output:

calculator			
0	1	2	3
4	5	6	7
8	9	+	-
*	/	=	c

Java source code:

```
package omerfanwe;
class callme {
    void call(String msg) {
        System.out.print("[ " + msg);
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {
            String msg; callme target; Thread t;
            System.out.println("Interrupted");
            System.out.print(" ]");
        }
    }
}

class caller implements Runnable {
    String msg; callme target; Thread t;
    public caller (callme targ, String s) {
        target = targ;
        msg = s;
    }
    t = new Thread (this)
    t.start ();
}

public void run () {
    t.call(target,call(msg));
    msg
}
```

## Class synch of

```
public static void main(String args[])
{
    CallMe target = new CallMe();
    Caller caller1 = new Caller(target, "Hello");
    Caller caller2 = new Caller(target, "Synchronized");
    Caller caller3 = new Caller(target, "World");
}
```

```
try {
    obj1.t.join();
    obj2.t.join();
    obj3.t.join();
} catch (InterruptedException e) {
    System.out.println("Interrupted");
}
```

Output:

Synchronized [world [Hello]]

1

三

Java source code:-

Server.java

Package OmanFauk;

import java.net.\*;

public class server {

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

ServerSocket s = new ServerSocket(2092);

System.out.println("Waiting for connections...");

Socket s1 = s.accept();

Socket

InputStream s1in = s1.getInputStream();

DataInputStream dis = new DataInputStream(s1in);

String receivedString = new (dis.readUTF());

System.out.printIn ("Received:" + received);

System.out.printIn ("Received:" + receivedString);

Output Stream s1out = s1.getOutputStream();

DataOutputStream dos = new DataOutputStream

(s1.getOutputStream());

(s1.getOutputStream().toUppercase());

dos.writeUTF(receivedString.toUpperCase());

dos.close();

s1out.close();

s1.close(); } }

### client.java

```
package e.omanfaruk;  
import java.net.*;  
import java.io.*;  
import java.util.Scanner;  
public class Client {  
    public static void main(String args[]) throws  
        IOException {  
        Socket s1 = new Socket("localhost", 2992);  
        OutputStream s1out = s1.getOutputStream();  
        DataOutputStream dos = new DataOutputStream  
            (s1out);  
        System.out.println("Write your message...");  
        Scanner sc = new Scanner(System.in);  
        String msg = sc.nextLine();  
        dos.writeUTF(msg);  
        InputStream s1in = s1.getInputStream();  
        DataInputStream dis = new DataInputStream  
            (s1in);  
        System.out.println("Output!");  
        System.out.println(dis.readUTF());  
        dis.close();  
        s1out.close();  
        s1.close();  
    }  
}
```

## Java Source Code

package OmanFawlk

java. import java.io.\*;

import java.net.\*;

public class UDPserver {

public static void main(String args[]) throws

DatagramSocket = new DatagramSocket(9876);

byte[] receiveData = new byte[1024];

byte[] sendData = new byte[1024];

while(true){

DatagramSocket serverSocket = new DatagramSocket();

(receiveData, receiveData.length);

System.out.println("Waiting for data...");

serverSocket.receive(receivePacket);

System.out.println("Data Received...");

Thread.sleep(3000);

InetAddress IPAddress = receivePacket.getSocketAddress();

int port = receivePacket.getPort();

String capitalization = sentence.toUpperCase();

SendData = capitalization.getBytes();

}

## UDPServer.java    UDPCClient.java

package OmanFawlk

```
import java.io.*;
public class UDPServer {
    public static void main(String args[]) throws Exception {
        DatagramSocket serverSocket = new DatagramSocket(9876);
        byte[] receiveData = new byte[1024];
        byte[] sendData = new byte[1024];
        while (true) {
            DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length);
            packet(receiveData, receiveData.length);
            byte[] sendData = new byte[1024];
            byte[] receiveData = new byte[1024];
            String sentence = inputUser.readLine();
            String sentence = sentence.getBytes();
            sendData = sentence.getBytes();
            DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, IPAddress, 9876);
            packet(sendData, sendData.length, IPAddress, 9876);
            System.out.println("Data Received ... ");
            System.out.println("From Server: " + modifiedSentence);
            clientSocket.close();
        }
    }
}
```