JAVALAB, II Bsc CS

EX1: IMPLEMENTING PACKAGE, INHERITANCE AND INTERFACE

```
import java.io.*;
import java.util.*;
public interface ex1
{
static float pi=3.14F;
float area(float rad);
float perim(float rad);
}
class circle implements ex1
{
public float area(float rad)
{
return(pi*rad*rad);
}
public float perim(float rad)
{
return(2*pi*rad);
}
}
class testinterface
{
public static void main(String args[])
{
circle c=new circle();
System.out.print("get radius");
Scanner in=new Scanner(System.in);
int r;
r=in.nextInt();
System.out.println("area of circle="+c.area(r));
System.out.println("perimeter of circle="+c.perim(r));
}
}
```

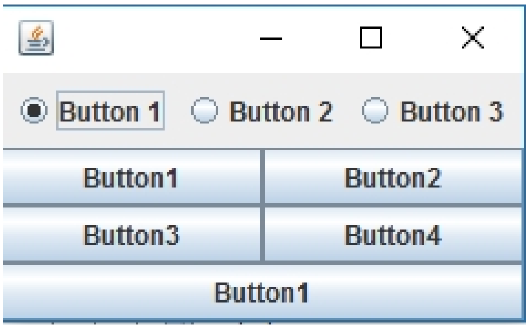
OUTPUT

```
GET RADIUS : 10
AREA OF CIRCLE : 314.0
PERIMETER OF CIRCLE: 62.800003
```

EX2: IMPLEMENTING FLOW, BORDER, & GRID LAYOUT

```
import javax.swing.JApplet;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class layout extends JApplet {
public static void main(String[]args){
JFrame frame=new JFrame();
JPanel radioPanel=new JPanel(new FlowLayout(FlowLayout.CENTER));
JPanel GridPanel=new JPanel(new GridLayout(2,2));
JPanel BorderPanel=new JPanel(new BorderLayout());
JRadioButton button1=new JRadioButton("Button 1",true);
JRadioButton button2=new JRadioButton("Button 2",false);
JRadioButton button3=new JRadioButton("Button 3",false);
radioPanel.add(button1);
radioPanel.add(button2);
radioPanel.add(button3);
GridPanel.add(new JButton("Button1"));
GridPanel.add(new JButton("Button2"));
GridPanel.add(new JButton("Button3"));
GridPanel.add(new JButton("Button4"));
BorderPanel.add(new JButton("Button1"));
frame.add(radioPanel,BorderLayout.NORTH);
frame.add(GridPanel,BorderLayout.CENTER);
frame.add(BorderPanel,BorderLayout.SOUTH);
frame.pack();
frame.setVisible(true);}
}
```

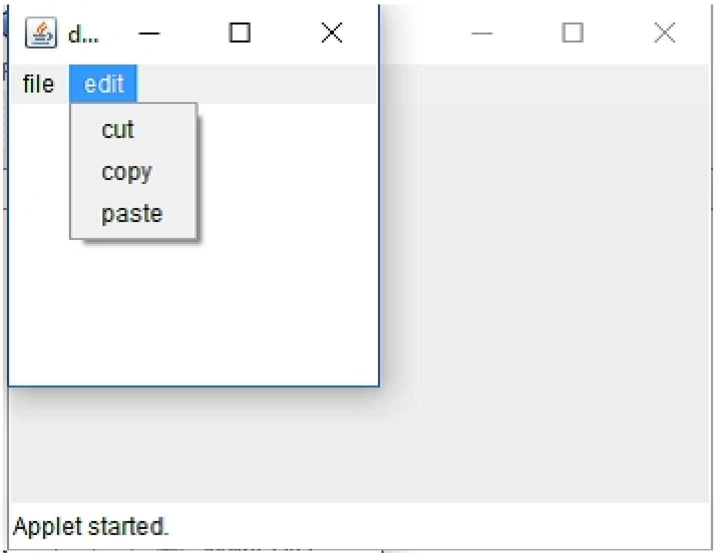
OUTPUT



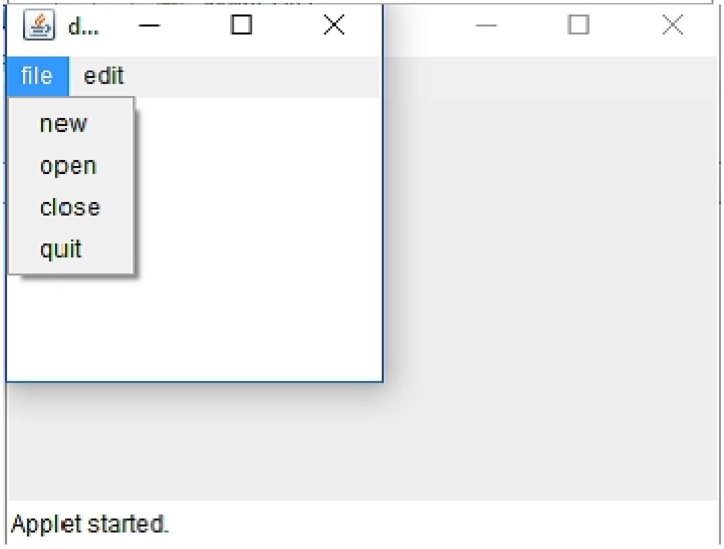
EX3: IMPLEMENTING DIALOGS, MENU & FRAME

```
import javax.swing.JApplet;
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class Dialog extends JApplet implements ActionListener {
Frame f;
MenuItem mi1;
public void init(){
f=new Frame("demo Frame");
f.setSize(200,200);
MenuBar mb=new MenuBar();
f.setMenuBar(mb);
Menu file=new Menu("file");
mi1=new MenuItem("new");
file.add(mi1);
mi1.addActionListener(this);
file.add(new MenuItem("open"));
file.add(new MenuItem("close"));
file.add(new MenuItem("quit"));
mb.add(file);
Menu Edit=new Menu("edit");
Edit.add(new MenuItem("cut"));
Edit.add(new MenuItem("copy"));
Edit.add(new MenuItem("paste"));
mb.add(Edit);
f.show();}
```

OUTPUT



Applet started.



Applet started.

EX4: IMPLEMENTING USER DEFINED EXCEPTION HANDLING

```
import java.io.*;
import java.util.*;
class OwnException extends Exception
{
OwnException()
{
}}
public class UserException
{
public static void main(String[] args)
{
System.out.print("Get voter's age");
Scanner in=new Scanner(System.in);
int age;
age=in.nextInt();
try
{
if(age<18)
throw new OwnException();
else
System.out.println("voter's Age is accepted");
}
catch(OwnException exp)
{
System.out.println("voter's age should be above 10");
System.out.println(exp);
}}}
```

OUTPUT

```
GET VOTER’S AGE
13
VOTER’S AGE SHOULD BE ABOVE 18
OWN EXCEPTION
GET VOTER’S AGE
45
VOTER’S AGE IS ACCEPTED
```

EX5: IMPLEMENTING MULTI-THREADING

```
class share extends Thread
{
static String msg[]={"this","is","a","synchronized","variable"};
share(String threadname)
{
super(threadname);
}
public void run()
{
display(getName());
}
public synchronized void display(String threadN)
{
for(int i=0;i<=4;i++)
System.out.println(threadN+msg[i]);
try
{
this.sleep(8000);
}
catch(Exception e)
{
} } }
public class multithreading
{
public static void main(String[]args)
{
share t1=new share("ThreadOne:");
t1.start();
share t2=new share("ThreadTwo:");
t2.start();
share t3=new share("ThreadThree:");
t3.start();
}
}
```

OUTPUT

```
Thread One :this
Thread One:is
Thread One:a
Thread One:synchronized
Thread One:Variable
Thread Three:this
Thread Two:this
Thread Three:is
Thread Three:a
Thread Three:synchronized
Thread Two:is
Thread Three:Variable
Thread Two:a
Thread Two:synchronized
Thread Two;Variable
```

EX6. IMPLEMENTING I/O STREAM FILE HANDLING

```
import java.io.*;
import java.util.*;
public class ex9
{
static Scanner in=new Scanner(System.in);
public static void main(String args[])throws IOException
{
DataOutputStream dos=new DataOutputStream(new FileOutputStream("test.dat"));
System.out.println("how many students?");
int no=in.nextInt();
for(int i=1;i<=no;i++)
{
System.out.print("enter name :");
String name=in.next();
System.out.print("enter role no :");
int role =in.nextInt();
System.out.print("enter the mark1 :");
int m1=in.nextInt();
System.out.print("enter the mark2 :");
int m2=in.nextInt();
dos.writeUTF(name);
dos.writeInt(role);
dos.writeInt(m1);
dos.writeInt(m2);
}
dos.close();
DataInputStream dis=new DataInputStream(new FileInputStream("test.dat")); for(int
i=1;i<=no;i++)
{
String n=dis.readUTF();
int role =dis.readInt();
int mark1=dis.readInt();
int mark2=dis.readInt();
int mark3=mark1+mark2;
System.out.println("===================");
System.out.println("name :"+n);
System.out.println("role no :"+role);
System.out.println("mark1 :"+mark1);
System.out.println("mark2 :"+mark2);
System.out.println("total :"+mark3);
System.out.println("===================");
}
dis.close();
}
}
```

OUTPUT

```
HOW MANY STUDENTS?
3
ENTER NAME : VISHNU
ENTER ROLL NUMBER : 1001
ENTER MARK1 : 98
ENTER MARK2 : 98
ENTER NAME : SOWNDHARYA
ENTER ROLL NUMBER : 1002
ENTER MARK1 : 99
ENTER MARK2 : 99
ENTER NAME : THAMOTHARAN
ENTER ROLL NUMBER : 1003
ENTER MARK1 : 100
ENTER MARK2 : 99
NAME : VISHNU
ROLL NUMBER : 1001
MARK 1 : 98
MARK 2 : 98
TOTAL : 196
=====================================================
NAME : SOWNDHARYA
ROLL NUMBER : 1002
MARK 1 : 99
MARK 2 : 99
TOTAL : 198
======================================================
NAME : THAMOTHARAN
ROLL NUMBER : 1003
MARK 1 :100
MARK 2 : 99
TOTAL : 199
=======================================================
```

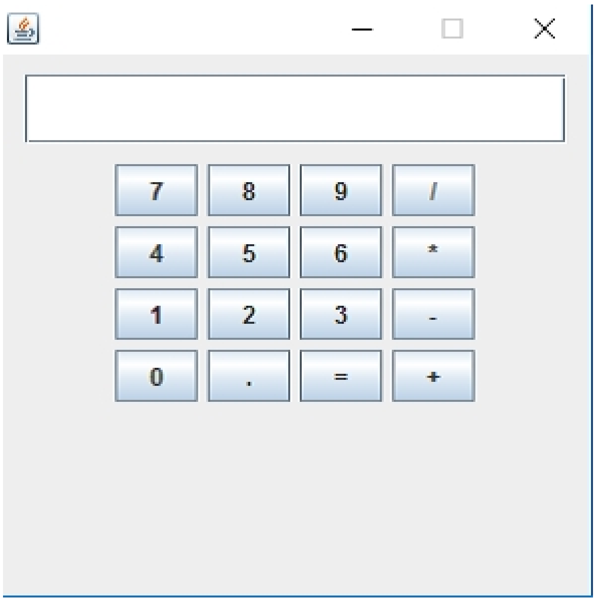
EX7: IMPLEMENTING CALCULATOR USING SWING

```
import javax.swing.JApplet;
import javax.swing.*;
import javax.swing.border.*;
import java.awt.*;
import java.awt.event.*;
import javax.script.ScriptEngineManager;
import javax.script.ScriptEngine;
import javax.script.ScriptException;
public class calcul extends JFrame implements ActionListener {
private JPanel buttonPanel;
private JPanel inputOutputPanel;
private JTextField info;
private boolean editable=true;
public calcul()
{
Container contentPane=getContentPane();
contentPane.setLayout(new FlowLayout());
buttonPanel=new JPanel();
buttonPanel.setLayout(new GridLayout(4,4,5,5));
inputOutputPanel=new JPanel();
inputOutputPanel.setLayout(new FlowLayout());
setSize(300,300);
setResizable(false);
info=new JTextField();
info.setPreferredSize(new Dimension(270,35));
info.addActionListener(this);
inputOutputPanel.add(info); contentPane .add(inputOutputPanel);
String buttons[]={"7","8","9","/","4","5","6","*","1","2","3","-","0",".","=","+",};
for(String i:buttons)
{
JButton button =new JButton(i);
button.addActionListener(this);
buttonPanel.add(button);
}
```

```
contentPane.add(buttonPanel);
}
public void actionPerformed(ActionEvent event)
{
if(event.getSource()instanceof JButton)
{
JButton clickedButton=(JButton)event.getSource();
if(clickedButton.getText().equals("="))
{
addOutput();
}
else
{
addInput(clickedButton.getText());
}
}
}
public void addInput(String line)
{
editable=true; info.setText(info.getText()+line);
}
public void addOutput()
{
double output=0;
try
{
ScriptEngineManager mgr=new ScriptEngineManager();
ScriptEngine engine=mgr.getEngineByName("javaScript");
info.setText(engine.eval(info.getText()).toString());
}
catch(ScriptException e)
{
info.setText("Syntax error");
}
editable=false;
```

```
}
public static void main(String[]args)
{
calcul calculator=new calcul();
calculator.setVisible(true);
}
}
```

OUTPUT



EX8: CRUD OPERATION USING JDBC

```
import java.sql.*;
public class CrudEx{
public static void main(String args[]){
try{
String url="jdbc:odbc:coll";
Class.forName("sun.jdbc.odbc.jdbcOdbcDriver");
Connection c=DriveManager.getConnection(url);
Statement st=c.createStatement();
ResultSet rs=st.executeQuery("select*from stud");
while(rs.next()){
System.out.println(rs.getString(1));
System.out.println(rs.getString(2));
System.out.println(rs.getString(3));
System.out.println("---------------------");
}
st.executeUpdate("insert into stud values(14,'sowndharya,467)");
st.executeUpdate("Delete from stud where stName='vishnu'");
st.executeUpdate("update stud set stName='malar'where i9d=11");
rs=st.executeOuery("select*from stud");
while(rs.next()){
System.out.println(rs.getString(1));
System.out.println(rs.getString(2));
System.out,println(rs.getString(3));
}}
catch(Exception ee){
System.out.println(ee);
}}}
```

OUTPUT

```
Init:
Run-single:
6
Sowndharya
567
11
Vishnu
- - - - - - - -- - - - - - - - - -- - - -------- --- - --- -- -
11
VISHNU
467
14
Malar
467
```

9.Tcp-ipclient

```
import java.net.*;

import java.io.*;

class tcpip_client
{
public static void main(String args[]) throws IOException
{
Socket s=null;
BufferedReader b=null;
try
{
s=new Socket(InetAddress.getLocalHost(),98);
b=new BufferedReader(new InputStreamReader(s.getInputStream()));

}
catch(UnknownHostException u)
{
System.err.println("I don't know host");
System.exit(0);
}
String imp;
while((imp=b.readLine())!=null)
{
System.out.println(imp);
}
b.close();
s.close();
}
}
```

OUTPUT

```
WELCOME TO JAVA

TCP and UDP Socket
```

9. Tcpip- server

```
import java.net.*;
import java.io.*;
class tcpip_server
{
public static void main(String args[])throws IOException
{
ServerSocket n1=null;
try
{
n1=new ServerSocket(98);
}
catch(IOException e)
{
System.err.println("port 98 could not be found");
System.exit(1);
}
Socket c=null;
try
{
c=n1.accept();
System.out.println("connection from"+c);
}
catch(IOException e)
{
System.out.println("accept failed");
System.exit(1);
}
PrintWriter out =new PrintWriter(c.getOutputStream(),true);
BufferedReader in=new BufferedReader(new InputStreamReader(c.getInputStream()));
String n;
BufferedReader sin=new BufferedReader(new InputStreamReader(System.in));
while((n=sin.readLine())!=null)
{
out.println(n);
}
out.close();
c.close();
n1.close();
} }
```

OUTPUT

```
WELCOME TO JAVA
TCP and UDP Socket
```

9.UDP-Client

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

class UDPClient
{
public static void main(String args[])throws Exception
{
BufferedReader inFromUser=new BufferedReader(new InputStreamReader
(System.in));
DatagramSocket clientSocket=new DatagramSocket();
InetAddress IPAddress=InetAddress.getByName("localhost");

byte[] sendData=new byte[1024];

byte[] receiveData=new byte[1024];

String sentence=inFromUser.readLine();

sendData=sentence.getBytes();

DatagramPacket sendPacket=new DatagramPacket

(sendData,sendData.length,IPAddress,9876);

clientSocket.send(sendPacket);

DatagramPacket receivePacket=new DatagramPacket

(receiveData,receiveData.length);

clientSocket.receive(receivePacket);

String modifiedSentence=new String(receivePacket.getData());

System.out.println("FROM SERVER:"+modifiedSentence);

clientSocket.close();

}

}
```

OUTPUT

```
FROM SERVER:WELCOME TO JAVA
```

9.UDP-Server

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

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);

serverSocket.receive(receivePacket);

String sentence=new String(receivePacket.getData());

System.out.println("RECEIVED:"+sentence);

InetAddress IPAddress=receivePacket.getAddress();

int port=receivePacket.getPort();

String capitalizedSentence=sentence.toUpperCase();

sendData=capitalizedSentence.getBytes();

DatagramPacket sendPacket=new DatagramPacket

(sendData,sendData.length,IPAddress,port);

serverSocket.send(sendPacket);

}

}

}
```

OUTPUT

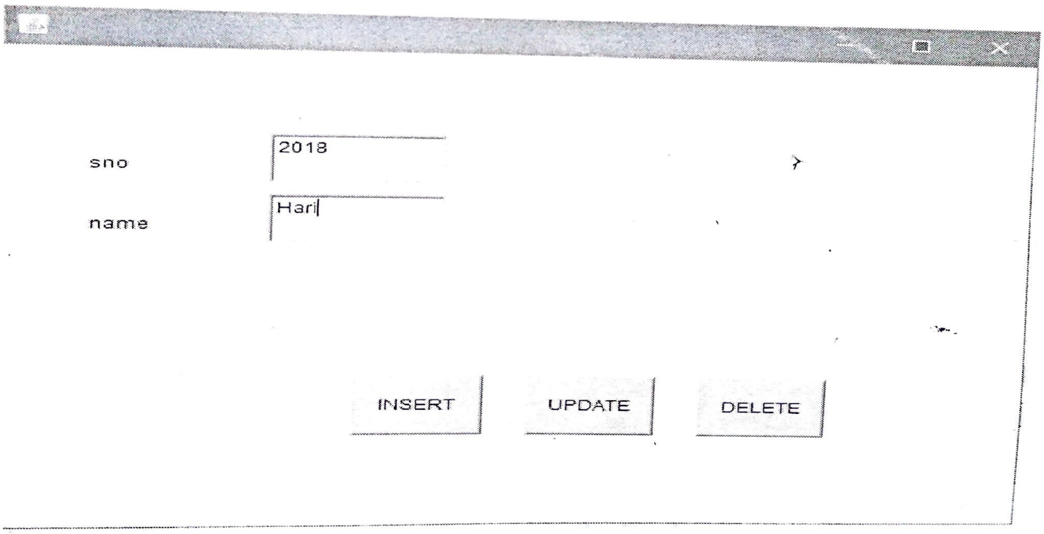
```
RECEIVED: Welcome To Java
```

10. GUI OPERATION

```
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
public class Ins extends Frame implements ActionListener
{
Frame f;
Label l1,l2;
TextField t1,t2;
Button b1,b2,b3,b4;
Connection c;
Statement s;
ResultSet r;
Ins()
{
try
{
f=new Frame();
f.setLayout(null);
f.setVisible(true);
f.setSize(100,600);
l1=new Label("sno");
l1.setBounds(50,100,100,50);
f.add(l1);
l2=new Label("name");
l2.setBounds(50,150,100,50);
f.add(l2);
t1=new TextField();
t1.setBounds(150,100,100,40);
f.add(t1);
t2=new TextField();
t2.setBounds(150,150,100,40);
f.add(t2);
b1=new Button("INSERT");
b1.setBounds(200,300,75,50);
f.add(b1); b1.addActionListener(this);
b2=new Button("UPDATE");
b2.setBounds(300,300,75,50);
f.add(b2);
b2.addActionListener(this);
b3=new Button("DELETE");
b3.setBounds(400,300,75,50);
f.add(b3);
 b3.addActionListener(this);
b4=new Button("EXIT");
b4.setBounds(200,300,75,50);
f.add(b4);
 b4.addActionListener(this);
String driverName="oracle.jdbc.driver.OracleDriver";
String url="jdbc:oracle:thin:@kolanji:1521:kolanji";
String user="system";
String password="manager";
Class.forName(driverName);
Connection c=DriverManager.getConnection(url,user,password);
s=c.createStatement();
}
catch(Exception e)
{
} }
public void actionPerformed(ActionEvent ae)
{
try
{
if(ae.getSource()==b1)
{
String s1="insert into staff(sno,name)values
(' "+t1.getText()+" ',' "+t2.getText()+" ')";
System.out.println(s1);
s.executeUpdate(s1);
r=s.executeQuery("select* from staff");
t1.setText(" ");
t2.setText(" ");
}
else if(ae.getSource()==b2)
{
String s2="update staff set name=' "+t2.getText()+" ' where
 sno= ' "+t1.getText()+" ' ";
System.out.println(s2);
s.executeUpdate(s2);
r=s.executeQuery("select * from staff");
t1.setText(" ");
t2.setText(" ");
}
else if(ae.getSource()==b3)
{
String s3="delete from staff where sno=' "+t1.getText()+" ' ";
System.out.println(s3);
s.executeUpdate(s3);
r=s.executeQuery("select * from staff");
t1.setText("");
t2.setText("");
}
else if(ae.getSource()==b4)
{
c.close();
f.dispose();
}
}
catch(Exception e)
{}}
public static void main(String args[])
{
new Ins();
```

```
}
  }
```

OUTPUT



sno 2018

name Hari