List of experiments for AJP PR exam

1.Write a program to demonstrate status of key on an Applet window -KeyPressed, KeyReleased,

```
Import java.awt*;
Import java.applet*;
Import.java.event;
/*
<applet code =''keyboardDemo" width=500 height=300>
*/
public class keyboardDemo extends applet
implement keylistener
{
   String msg= " ";
   Public void init()
    {
      Add keylistener(this);
      Request focus();
    }
     Public void keypressed (keyevent k)
       {
          Showstatus("keypressed")
        }
    Public void keyreleased (keyevent k)
       {
         Showstatus("keyreleased")
       }
    Public void paint (graphics g)
      {
        g.drawstring(msg,30,70);
     }
}
```

OR

```
// importing awt libraries
import java.awt.*;
import java.awt.event.*;
// class which inherits Frame class and implements KeyListener interface
public class KeyListenerExample extends Frame implements KeyListener {
// creating object of Label class and TextArea class
Label l;
TextArea area;
// class constructor
KeyListenerExample() {
// creating the label
l = new Label();
```

```
// setting the location of the label in frame
l.setBounds (20, 50, 100, 20);
// creating the text area
area = new TextArea();
// setting the location of text area
area.setBounds (20, 80, 300, 300);
// adding the KeyListener to the text area
area.addKeyListener(this);
// adding the label and text area to the frame
add(l);
add(area);
// setting the size, layout and visibility of frame
setSize (400, 400);
setLayout (null);
setVisible (true);
}
// overriding the keyPressed() method of KeyListener interface where we set the text of the label whe
n key is pressed
public void keyPressed (KeyEvent e) {
l.setText ("Key Pressed");
}
// overriding the keyReleased() method of KeyListener interface where we set the text of the label w
hen key is released
public void keyReleased (KeyEvent e) {
l.setText ("Key Released");
}
// overriding the keyTyped() method of KeyListener interface where we set the text of the label when
a key is typed
public void keyTyped (KeyEvent e) {
l.setText ("Key Typed");
}
// main method
public static void main(String[] args) {
new KeyListenerExample();
}
}
```

2.Write a program to demonstrate status of key on an Applet window - KeyUp, KeyDown.

```
Import java.awt*;
Import java.applet*;
Import.java.event;
/*
<applet code =''keyboardDemo" width=500 height=300>
*/
public class keyboardDemo extends applet
implement keylistener
{
   String msg= " ";
```

```
   Public void init()
    {
      Add keylistener(this);
      Request focus();
    }
     Public void keyup (keyevent k)
       {
          Showstatus(“keyup”)
        }
    Public void keydown (keyevent k)
       {
         Showstatus(“keydown”)
       }
    Public void paint (graphics g)
      {
        g.drawstring(msg,30,70);
     }
}
```

3. Write a program to create a frame using AWT. Implement mouseClicked, Frame should become visible when the mouse enters it.

```
import java.awt.*;
import java.awt.event.*;
public class MouseListener extends Frame implements MouseListener{
Label l;
MouseListener(){
addMouseListener(this);
l=new Label();
l.setBounds(20,50,100,20);
add(l);
setSize(300,300);
setLayout(null);
setVisible(true);
}
public void mouseClicked(MouseEvent e) {
l.setText("Mouse Clicked");
}
public void mouseEntered(MouseEvent e) {
l.setText("Mouse Entered");
}
public void mouseExited(MouseEvent e) {
l.setText("Mouse Exited");
}
public void mousePressed(MouseEvent e) {
l.setText("Mouse Pressed");
}
```

```
public void mouseReleased(MouseEvent e) {
l.setText("Mouse Released");
}
public static void main(String[] args) {
new MouseListenerExample();
}
}
```

4. Write a program to create a frame using AWT. Implement mouseEntered() and mouseExited() events.

```
 import java.awt.*;
import java.awt.event.*;
public class MouseListener extends Frame implements MouseListener{
Label l;
MouseListener(){
addMouseListener(this);
l=new Label();
l.setBounds(20,50,100,20);
add(l);
setSize(300,300);
setLayout(null);
setVisible(true);
}
public void mouseClicked(MouseEvent e) {
l.setText("Mouse Clicked");
}
public void mouseEntered(MouseEvent e) {
l.setText("Mouse Entered");
}
public void mouseExited(MouseEvent e) {
l.setText("Mouse Exited");
}
public void mousePressed(MouseEvent e) {
l.setText("Mouse Pressed");
}
public void mouseReleased(MouseEvent e) {
l.setText("Mouse Released");
}
public static void main(String[] args) {
new MouseListenerExample();
}
}
```

5. Develop a GUI which accepts the information regarding the marks for all the subjects of a student in the examination. Display the result for a student in a separate window.

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.JApplet;
import javax.swing.*;
/**
*
* @author prajakta
*/
public class NewJApplet extends JApplet implements ActionListener{
/**
* Initialization method that will be called after the applet is loaded into
* the browser.
*/
JTextField T1,T2,T3,T4,T5,ans;
JLabel L1, L2, L3, L4, L5;
JButton res;
@Override
public void init() {
// TODO start asynchronous download of heavy resources
// JFrame f = new JFrame();
Container contentPane = getContentPane();
contentPane.setLayout(new FlowLayout(FlowLayout.CENTER));
res = new JButton("Result");
T1 = new JTextField(20);
T2 = new JTextField(20);
T3 = new JTextField(20);
T4 = new JTextField(20);
T5 = new JTextField(20);
ans = new JTextField(20);
L1 = new JLabel("Enter marks of AJP");
L2 = new JLabel("Enter marks of PDC");
L3 = new JLabel("Enter marks of CN");
L4 = new JLabel("Enter marks of OOP");
L5 = new JLabel("Enter marks of PM");
contentPane.add(L1);
contentPane.add(T1);
contentPane.add(L2);
contentPane.add(T2);
contentPane.add(L3);
contentPane.add(T3);
contentPane.add(L4);
contentPane.add(T4);
contentPane.add(L5);
contentPane.add(T5);
contentPane.add(res);
contentPane.add(ans);
```

```
res.addActionListener(this);
}
@Override
public void actionPerformed(ActionEvent e) {
int m1 =Integer.parseInt(T1.getText());
int m2 = Integer.parseInt(T2.getText());
int m3 = Integer.parseInt(T3.getText());
int m4 = Integer.parseInt(T4.getText());
int m5 = Integer.parseInt(T5.getText());
double result = ((m1+m2+m3+m4+m5)/500.0)*100;
String format = String.format("%.2f", result);
ans.setText(format);
}
}
```

6. Develop a GUI which accepts the information regarding personal details of a student. Display the information of student in a dialog box.

```
import java.sql.*;
public class MySQLdatabase {
public static void main(String[] args) {
try {
Class.forName("com.mysql.cj.jdbc.Driver");
Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/manisha",
"root", "Root2223");
Statement s = con.createStatement();
s.execute("create table student ( stud_id integer,stud_name varchar(20),stud_address
varchar(30) )"); // create a table
s.execute("insert into student values(001,'ARman','Delhi')"); // insert first row into the
table
table
table
s.execute("insert into student values(002,'Robert','Canada')"); // insert second row into the
s.execute("insert into student values(003,'Ahuja','Karnal')"); // insert third row into the
ResultSet rs = s.executeQuery("select * from student");
if (rs != null) // if rs == null, then there is no record in ResultSet to show
while (rs.next()) // By this line we will step through our data row-by-row
{
System.out.println(" ");
System.out.println("Id of the student: " + rs.getString(1));
System.out.println("Name of student: " + rs.getString(2));
System.out.println("Address of student: " + rs.getString(3));
System.out.println(" ");
}
s.close(); // close the Statement to let the database know we're done with it
con.close(); // close the Connection to let the database know we're done with it
} catch (SQLException err) {
System.out.println("ERROR: " + err);
} catch (Exception err) {
```

```
System.out.println("ERROR: " + err);
}
}
}
```

7. Write a program to demonstrate the use of InetAddress class and its factory methods.

```
import java.io.*;
import java.net.*;
import java.util.*;
class GFG {
public static void main(String[] args)
throws UnknownHostException
{
// To get and print InetAddress of Local Host
InetAddress address1 = InetAddress.getLocalHost();
System.out.println("InetAddress of Local Host : "
+ address1);
// To get and print InetAddress of Named Host
InetAddress address2
= InetAddress.getByName("45.22.30.39");
System.out.println("InetAddress of Named Host : "
+ address2);
// To get and print ALL InetAddresses of Named Host
InetAddress address3[]
= InetAddress.getAllByName("172.19.25.29");
for (int i = 0; i < address3.length; i++) {
System.out.println(
"ALL InetAddresses of Named Host : "
+ address3[i]);
}
// To get and print InetAddresses of
// Host with specified IP Address
byte IPAddress[] = { 125, 0, 0, 1 };
InetAddress address4
= InetAddress.getByAddress(IPAddress);
System.out.println(
"InetAddresses of Host with specified IP Address : "
+ address4);
// To get and print InetAddresses of Host
// with specified IP Address and hostname
byte[] IPAddress2
= { 105, 22, (byte)223, (byte)186 };
InetAddress address5 = InetAddress.getByAddress(
"gfg.com", IPAddress2);
System.out.println(
"InetAddresses of Host with specified IP Address and hostname : "
+ address5);
}
```

}

8. Write a simple JSP page to display a simple message .

```
<html>
        <head>
                <title>my first JSP</title>
                <meta charset=”UTF-8”<
                <meta name=”viewport” content=’’width=device-width, intitial-scale=1.0”>
        </head>
                <body>
                        <div>Hello All !</div>
                </body>
</html>
```