Experiment 1

```
package practicals;

/*
<applet code="KeyboardDemo" width="300" height="300">
</applet>
*/

import java.awt.*;
import java.awt.event.*;
import java.applet.*;

public class Exp1_Key extends Applet implements KeyListener
{

	//String msg = "";

	public void init()
	{
		addKeyListener(this);
		requestFocus();
	}
	public void keyPressed (KeyEvent e)
	{
		showStatus ("Key Pressed");
		repaint();
	}
	 public void keyReleased (KeyEvent e)
	 {
	 showStatus ("Key Released");
	 repaint();
    }
	 public void keyTyped (KeyEvent e) {
	 showStatus ("Key Typed");
	 repaint();
    }

	 /*public void paint(Graphics g)
	 {
		g.drawString(msg, 30, 70);
	 }*/

}
```

Experiment 2

```
package practicals;


import java.awt.*;
import java.awt.event.*;
public class Exp2_Mouse extends Frame implements MouseListener{
   Label l;
   Exp2_Mouse(){
      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 Exp2_Mouse();
}
}
```

Experiment 3

```
package practicals;

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.io.*;

public class Exp3_Marks {

   // Function to write a student information in JFrame and storing it in a file
   public static void StudentInfo()
   {
        // Creating a new frame using JFrame
      JFrame f = new JFrame("Student Grade Calculator");

      // Creating the labels
      JLabel l1, l2, l3, l4, l5,l6, l7, l8;

      // Creating three text fields for student name, college mail ID and for  Mobile No
      JTextField t1, t2, t3, t4, t5, t6, t7, t8;

      // Creating two JComboboxes for Branch and for Section
     //  JComboBox j1, j2;

      // Creating  two buttons
      JButton b1, b2;

      // Naming the labels and setting
      // the bounds for the labels
```

```
l1 = new JLabel("Student Name:");
l1.setBounds(50, 50, 450, 30);
l2 = new JLabel("Enter Marks of Subject 1");
l2.setBounds(50, 90, 450, 30);
l3 = new JLabel("Enter Marks of Subject 2");
l3.setBounds(50, 120, 450, 30);
l4 = new JLabel("Enter Marks of Subject 3");
l4.setBounds(50, 150, 450, 30);
l5 = new JLabel("Enter Marks of Subject 4");
l5.setBounds(50, 180, 450, 30);


b1 = new JButton("Calculate Total and Grade");
b1.setBounds(50, 230, 450, 30);

l6 = new JLabel("Total");
l6.setBounds(50, 270, 250, 30);
l7 = new JLabel("Percentage");
l7.setBounds(50, 300, 450, 30);
l8 = new JLabel("Grade");
l8.setBounds(50, 330, 450, 30);

// Creating the textfields and
// setting the bounds for textfields
t1 = new JTextField();
t1.setBounds(300, 50, 170, 30);
t2 = new JTextField();
t2.setBounds(300, 90, 70, 30);
t3 = new JTextField();
t3.setBounds(300, 120, 70, 30);
t4 = new JTextField();
```

```
t4.setBounds(300, 150, 70, 30);
t5 = new JTextField();
t5.setBounds(300, 180, 70, 30);

t6 = new JTextField();
t6.setBounds(300, 270, 70, 30);
t7 = new JTextField();
t7.setBounds(300, 300, 70, 30);
t8 = new JTextField();
t8.setBounds(300, 330, 70, 30);



// Creating one button for Saving and other button to close
// and setting the bounds

b2 = new JButton("Close");
b2.setBounds(450, 350, 70, 30);




// Adding action listener
b1.addActionListener(new ActionListener() {
   public void actionPerformed(ActionEvent e)
   {

      // Getting the text from text fields
      // and JComboboxes
      // and copying it to a strings



      if (e.getSource() == b1) {
```

```
try {

      int i2 = Integer.parseInt(t2.getText());
   int i3 = Integer.parseInt(t3.getText());
   int i4 = Integer.parseInt(t4.getText());
   int i5 = Integer.parseInt(t5.getText());

      double total = i2+i3+i4+i5;
  t6.setText("" + total);

  // Calculates percentage
  double per = (total / 400) * 100;
  t7.setText("" + per);
  String grade = null;
  // Grade calculation
  if (per > 90)
  { grade = "A+";
  } else if ((per > 85) && (per < 90)) {
        grade = "A";
  } else if ((per > 80) && (per < 85)) {
        grade = "B+";
  } else if ((per > 70) && (per < 80)) {
        grade = "B";
  } else if ((per > 60) && (per < 70)) {
        grade = "C+";
  } else if ((per > 50) && (per < 60)) {
        grade = "C";
  }
        else if ((per > 40) && (per < 50)) {
        grade = "D";
  }
```

```
                else if ((per > 00) && (per < 40)) {
                grade = "Fail";
            }
          // Displays result in TextField
            t8.setText("" + grade);
          }
          catch (Exception ae) {
             System.out.println(ae);
          }
       }


  }
});

// Action listener to close the form
b2.addActionListener(new ActionListener() {
   public void actionPerformed(ActionEvent e)
   {
      f.dispose();
   }
});

// Default method for closing the frame
f.addWindowListener(new WindowAdapter() {
   public void windowClosing(WindowEvent e)
   {
      System.exit(0);
   }
});
```

```
        // Adding the created objects
        // to the frame
        f.add(l1);
        f.add(t1);
        f.add(l2);
        f.add(t2);
        f.add(l3);
        f.add(t3);
        f.add(l4);
        f.add(t4);
        f.add(l5);
        f.add(t5);
        f.add(l6);
        f.add(t6);
        f.add(l7);
        f.add(t7);
        f.add(l8);
        f.add(t8);
        f.add(b1);
        f.add(b2);
        f.setLayout(null);
        f.setSize(1700, 1600);
        f.setVisible(true);
    }
    // Driver code
    public static void main(String args[])
    {
        StudentInfo();
    }
}
```

Experiment 4

```
package practicals;

import java.sql.*;

public class Exp4_jdbc {

	public static void main(String[] args) {

		try{
			Class.forName("com.mysql.jdbc.Driver");
			Connection
con=DriverManager.getConnection("jdbc:mysql://localhost/mysqldatabase","root","123
4");
			Statement s=con.createStatement();
			s.execute("create table Students1(id int,name
varchar(20),rollno int)");
			s.execute("insert into Students1 values(1,'Gayatri',19)");
			s.execute("insert into Students1 values(2,'Kriti',34)");
			s.execute("insert into Students1 values(3,'Ansh',3)");
			s.execute("delete from Students1 where name='Gayatri'");
			ResultSet rs=s.executeQuery("select* from Students");
			if(rs!=null)
			while(rs.next()){
				System.out.println("----------------------------------
---");
				System.out.println("The Id is " + rs.getString(1));
				System.out.println("The Name is " + rs.getString(2));
				System.out.println("The Roll number is " +
rs.getString(3));
				System.out.println("----------------------------------
---");

			}
			con.close();
			s.close();
		}catch(Exception e){
			System.out.println("Error"+e);
		}


	}

}
```

Experiment 6 Inet

```
package practicals;


import java.io.*;
import java.net.*;
public class Exp6_Inet{
public static void main(String[] args){
try
	{
		InetAddress add1 = InetAddress.getLocalHost();
		System.out.println("InetAddress of Local Host: "+add1);

		//Actual:-
		InetAddress ip=InetAddress.getByName("www.pccoer.com");
		System.out.println("Host Name: "+ip.getHostName());
		System.out.println("IP Address: "+ip.getHostAddress());  //till here

		System.out.println("IP Address named host: "+ip);
		System.out.println("----------");
		InetAddress add2[]=InetAddress.getAllByName("192.168.137.1");
		for(int i=0; i<add2.length;i++){
			System.out.println("All InetAddresses of Named Host: "+add2);
		}
		byte IPAddress[]={125,0,0,1};
		InetAddress add3 = InetAddress.getByAddress(IPAddress);
		System.out.println("InetAddress of Host with specific IP Address: "+add3);

	}
	catch(Exception e)
	{
```

```
                System.out.println(e);
        }
}
}
```

Experiment 8 JDBC

```
package practicals;

import java.sql.*;

public class Exp8_jdbc {

	public static void main(String[] args) {

		try{
			Class.forName("com.mysql.jdbc.Driver");
			Connection
con=DriverManager.getConnection("jdbc:mysql://localhost/mysqldatabase","root","123
4");
			Statement s=con.createStatement();
			String sql = "create database students";
			s.executeUpdate(sql);
			}
		catch(Exception e){
			System.out.println(e);
		}


	}


}
```

EXPERIMENT 7 Servlet

LoginController.Java

```
package com.candid;


import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;



@WebServlet("/LoginController")

public class LoginController extends HttpServlet {


   protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

      String un = request.getParameter("username");

      String pw = request.getParameter("password");


      if (un.equals("admin") && pw.equals("admin")) {

         response.sendRedirect("success.html");

         return;

      } else {

         response.sendRedirect("error.html");

         return;

      }

   }


}
```

Login.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
    <title>Insert title here</title>
</head>
<body>
Sample login Example (try with username as "admin" and password as "admin" without
quart ) <br> <br>
<form action="LoginController" method="post">
    Enter username :<input type="text" name="username"> <br>
    Enter password :<input type="password" name="password"><br>
    <input type="submit" value="Login">
</form>

</body>
</html>
```

Error.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
Invalid username or password
</body>
</html>
```

Success.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
Login Successful
<br>
<br>
<center>
Welcome to PCCOER College
</center>
</body>
</html>
```

Experiment 9 Simple JSP

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>

   <center>
   Welcome to PCCOER
   <br>
   <br>
   <font color="gray" size="5">
   The date is: <%= new java.util.Date() %>
   </font>
   </center>

</body>
</html>
```

Experiment 10 Calculator

Calculator.java

```
import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;


@WebServlet("/Calculator")
public class Calculator extends HttpServlet {
	private static final long serialVersionUID = 1L;

	protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
		response.setContentType("text/html");
		PrintWriter out=null;
		try {

			out=response.getWriter();
			out.println("<center>");
			int a=Integer.parseInt(request.getParameter("t1"));
			int b=Integer.parseInt(request.getParameter("t2"));
			int c=0;
			String op=request.getParameter("btn");
			if(op.equals("+")) c=a+b;
			if(op.equals("-")) c=a-b;
			if(op.equals("*")) c=a*b;
			if(op.equals("/")) c=a/b;
			out.println("<h3>"+a+op+b+" = "+c+"</h3>");


		}catch(Exception e) {

			out.println("Error:"+e.getMessage());

		}

		finally {
			out.println("<br>");
			out.println("To go to main page <a href=index.html> Click HERE
</a>");
			out.println("</center>");
		}

	}


	protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {


	}
```

```
}
```

index.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Calculator</title>
</head>
<body>
<center>
<form method= get action=Calculator>

Enter first number <input type = "text" id="t1" name="t1"/><br/>

Enter second number<input type = "text" id="t2" name="t2"/><br/>

<input type = "Submit" value="+" name="btn"/>
<input type = "Submit" value="-" name="btn"/>
<input type = "Submit" value="*" name="btn"/>
<input type = "Submit" value="/" name="btn"/>

</form>
</center>
</body>
</html>
```

Experiment 5 Palindrome

Palinclient.java

```
import java.rmi.*;

public class palin_client {

public static void main(String args[])

{

String value="PCCOER";

boolean answer;

try

{

// lookup method to find reference of remote object

string_palindrome access =

(string_palindrome)Naming.lookup("rmi://localhost:5600"+"/PALINDROME");

answer = access.palin("PCCOER");

if (answer)

System.out.println("String is PALINDROME!!!");

else

System.out.println("String is NOT PALINDROME!");

}

catch(Exception ae)

{

System.out.println(ae);

}

}
}
```

Palinimpl.java

```
//Java program to implement the palindrome interface

import java.rmi.*;

import java.rmi.server.*;

import java.rmi.server.UnicastRemoteObject;

/*******************UnicastRemoteObject*****************************

Used for exporting a remote object with JRMP and

obtaining a stub that communicates to the remote object

*****************************************************************/

public class palin_impl extends UnicastRemoteObject implements
string_palindrome{

// Default constructor to throw RemoteException and create object for
UnicastRemoteObjec

// from its parent constructor

palin_impl() throws RemoteException

{

super();

}

// Implementation of the palin interface

public boolean palin(String str) throws RemoteException{


// Pointers pointing to the beginning

// and the end of the string

int i = 0, j = str.length() - 1;

// While there are characters to compare

while (i < j) {

// If there is a mismatch

if (str.charAt(i) != str.charAt(j))

return false;
```

```
// Increment first pointer and
// decrement the other
i++;
j--;
}
// Given string is a palindrome
return true;
}
}
```

Palinserver.java

```
// Java program for server application

import java.rmi.*;

import java.rmi.registry.*;

public class palin_server {

public static void main(String args[])

{

try

{

// Create an object of the interface string_palindrome

// implementation class palin_impl
```

```
string_palindrome obj1 = new palin_impl();

// rmi registry within the server JVM with

// port number 5600

LocateRegistry.createRegistry(5600);

// Binds the remote object by the name

Naming.rebind("rmi://localhost:5600"+"/PALINDROME",obj1);

/**************************************************************

void java.rmi.Naming.rebind(String name, Remote obj) throws
RemoteException,

Rebinds the specified name to a new remote object.

Any existingbinding for the name is replaced.

Parameters:name a name in URL format

(without the scheme component)obj new remote object to associate with the
name

**************************************************************/
System.err.println("Server is ready");
}
catch(Exception ae)
```

{

System.out.println(ae);

}

}

}

Stringpalindrome.java

//Step I

//Creating a String_palindrome interface

import java.rmi.*;

public interface string_palindrome extends Remote

{

// Declaring the method prototype

public boolean palin(String str) throws RemoteException;

}