# LAB MANUAL
# OF
# JAVA LABORATORY
# (5CSL02)

**for**
**V Semester,**
**Department of CSE**
**Siddaganga Institute of Technology**
**Tumkur 572103**

# 1. Classes and objects

**a. Write a program in Java with class Rectangle with the data fields width, length, area and color. The length, width, area are of double type and color is of string type. The methods are set_length (), set_width (), set_color (), and find_area (). Create two objects of Rectangle and compare their area and color. If area and color both are the same for the objects then display "Matching Rectangles", otherwise display "Non Matching Rectangles".**

**PROGRAM:**

```
import java.util.Scanner;

class Rectangle
{
    double width, length, area;
    String color;
    void set_length ()
    {
            Scanner ob= new Scanner(System.in);
            System.out.println("\n Enter the Length:");
            length=ob.nextDouble();
    }
    void set_width ()
    {
            Scanner ob= new Scanner(System.in);
            System.out.println("\n Enter the Width:");
            width=ob.nextDouble();
    }
    void set_color ()
    {
            Scanner ob= new Scanner(System.in);
            System.out.println("\n Enter the Color:");
            color=ob. next ();
    }
    double find_area ()
```

```
    {
        area=length*width;
        return area;
    }
    void compare (Rectangle ob)
    {
        if(area==ob.area && color.equals(ob.color))
        System.out.println("Matching Rectangle\n");
        else
        System.out.println("Non Matching Rectangles\n");
    }
}
public class rect
{
    public static void main (String args [])
    {
        double area;
        Rectangle r1=new Rectangle ();
        Rectangle r2=new Rectangle ();

        System.out.println("Enter the 1st Rectangle dimension\n");
        r1.set_length ();
        r1.set_width ();
        r1.set_color ();
        area=r1.find_area ();
        System.out.println("\n1st Rectangle Area:"+area);

        System.out.println("Enter the 2nd Rectangle dimension\n");
        r2.set_length ();
        r2.set_width ();
        r2.set_color ();
        area=r2.find_area ();
        System.out.println("\n2nd Rectangle Area:"+area);

        r1. compare(r2);
    }
}
```

## STEPS FOR EXECUTION:

1. Go to your workspace, Right-click->Open terminal here
2. Create a file named rect.java (vi rect.java)
3. Type the program given above and save it (esc+Shift+: wq!)
4. Compile the program using the command javac rect.java
5. Run the program using the command java rect

## OUTPUT:

Enter the 1st Rectangle dimension

Enter the Length

2

Enter the Width

4

Enter the Color

Red

1st Rectangle Area:8

Enter the 2nd Rectangle dimension

Enter the Length

3

Enter the Width

6

Enter the Color

Red

2nd Rectangle Area:18

Non Matching Rectangles

## b. **Write a java program to overload constructor and method.**

**PROGRAM:**

```
class Shape
{
	double lnt, bdt, a;
	Shape (double l)
	{
		lnt=l;
	}
	Shape (double l, double b)
	{
		lnt=l;
		bdt=b;
	}

	void area (double l)
	{
		a=l*l;
		System.out.println("Area of Rectangle (only length is given) ="+a);
	}
	void area (double l, double b)
	{
		a=l*b;
		System.out.println("Area of Rectangle is="+a);
			}
}
Public class overload
{
	public static void main (String args [])
	{
		Shape s1=new Shape (5.000);
		Shape s2=new Shape (5.000,6.000);

		s1.area(s1.lnt);
		s2. area (s2.lnt, s2.bdt);

		/*System.out.println("Now if we pass only the length for Rectangle");
		s2. area(s2.lnt);*/
	}
}
```

## STEPS FOR EXECUTION:

1. Go to your workspace, Right-click->Open terminal here
2. Create a file named overload.java (vi overload.java)
3. Type the program given above and save it (esc+Shift+: wq!)
4. Compile the program using the command javac overload.java
5. Run the program using the command java overload

## OUTPUT:

Area of Rectangle (only length is given) =5000.0

Area of Rectangle is=3.0E7

# 2) Inheritance and polymorphism.

## i) Write a program in Java to create a Player class. Inherit the classes Cricket _Player, Football _Player and Hockey_ Player from Player class

## PROGRAM:

```
class Player
{
	String name;
	int age, matches, ranking;
	Player (String n, int a, int m, int r)
	{
		name=n;
		age=a;
		matches=m;
		ranking=r;
	}
}
class Cricket_Player extends Player
{
	int hscore, baverage, bataverage;
	Cricket_Player (String n,int a,int m,int r,int hs,int ba,int balla)
	{
		super(n,a,m,r);
		hscore=hs;
```

```
            bataverage=ba;
            baverage=balla;
        }
        void disp ()
        {
            System.out.println("Name: "+name);
            System.out.println("Age: "+age);
            System.out.println("No. of Matches: "+matches);
            System.out.println("Highest Score: "+hscore);
            System.out.println("Batting Average: "+bataverage);
            System.out.println("Balling Average: "+baverage);
            System.out.println("Player Ranking: "+ranking);
        }
}
class Football_Player extends Player
{
        int goals, gavg, pass;
        Football_Player (String n,int a,int m,int r,int g,int gaverage, int passeff)
        {
            super(n,a,m,r);
            goals=g;
            gavg=gaverage;
            pass=passeff;
        }
        void disp ()
        {
            System.out.println("Name: "+name);
            System.out.println("Age: "+age);
```

```
                System.out.println("No. of Matches: "+matches+"\n");
                System.out.println("No. of Goals: "+goals);
                System.out.println("Goals Average: "+gavg);
                System.out.println("Passing Efficiency: "+pass+"%");
                System.out.println("Player Ranking: "+ranking);
        }
}
class Hockey_Player extends Player
{
        int goals, gavg, pass;
        Hockey_Player (String n,int a,int m,int r,int g,int gaverage, int passeff)
        {
                super(n,a,m,r);
                goals=g;
                gavg=gaverage;
                pass=passeff;
        }
        void disp()
        {
                System.out.println("Name: "+name);
                System.out.println("Age: "+age);
                System.out.println("No. of Matches: "+matches);
                System.out.println("No. of Goals: "+goals);
                System.out.println("Goals Average: "+gavg);
                System.out.println("Passing Efficiency: "+pass+"%");
                System.out.println("Player Ranking: "+ranking);
        }
}
```

```
public class Inheritance
{
    public static void main (String args[])
    {
        Cricket_Player C=new Cricket_Player ("Sachin Tendulkar”,
        38,600,8,200,55,60);
        Football_Player F=new Football_Player ("Sunil Chhetri",32,120,90,3,80,94);
        Hockey_Player H=new Hockey_Player ("Dhanraj Pillay",32,120,90,3,80,94);
        C.disp ();
        F. disp ();
        H. disp ();
    }
}
```

## STEPS FOR EXECUTION:

1. Go to your workspace, Right-click->Open terminal here
2. Create a file named Inheritance.java (vi Inheritance.java)
3. Type the program given above and save it (esc+Shift+: wq!)
4. Compile the program using the command javac Inheritance.java
5. Run the program using the command java Inheritance

## OUTPUT:

Name: Sachin Tendulkar

Age: 38

No of Matches: 600

Highest Score: 200

Batting Average: 55

Bowling Average: 60

Player Ranking: 8

Name: Lionel Messi

Age=32

No of Matches: 120

No of Goals: 3

Goals Average: 80

Passing Efficiency:94%

Player Ranking: 90

Name: Dhanraj Pillay

Age=32

No of Matches: 120

No of Goals: 3

Goals Average: 80

Passing Efficiency:94%

Player Ranking: 90

**ii) Consider the trunk calls of a telephone exchange. A trunk call can be ordinary, urgent or lightning. The charges depend on the duration and the type of the call. Write a program using the concept of polymorphism in Java to calculate the charges**

**PROGRAM:**

```
class TrunkCall
{
        int duration;
        TrunkCall (int sec)
        {
          duration=sec;
        }

        double charge ()
        {
         System.out.println("Charge is undefined");
         return 0;
        }
}

class Ordinary extends TrunkCall
{
        Ordinary (int a)
        {
         super(a);
        }
        double charge ()
        {
          return duration*1.00;
        }
}

class Urgent extends TrunkCall
{
        Urgent (int a)
        {
          super(a);
        }
        double charge ()
        {
          return duration*2.00;
        }
```

```
}

class Lightning extends TrunkCall
{
        Lightning (int a)
        {
         super(a);
        }
        double charge ()
        {
          return duration*3.00;
        }
}

public class trunk
{
        public static void main (String args [])
    {
        TrunkCall a;
        Ordinary b=new Ordinary (50);
        Urgent c=new Urgent (70);
        Lightning d=new Lightning (20);
        a=b;
        System.out.println("Charges for Ordinary call="+a. charge ());
        a=c;
        System.out.println("Charges for Urgent call="+a. charge ());
        a=d;
        System.out.println("Charges for Lightning call="+a. charge ());
    }
}
```

## STEPS FOR EXECUTION:

1. Go to your workspace, Right-click->Open terminal here
2. Create a file named trunk.java (vi trunk.java)
3. Type the program given above and save it (esc+Shift+: wq!)
4. Compile the program using the command javac trunk.java
5. Run the program using the command java trunk

## OUTPUT:

```
Charges for Ordinary call=50

Charges for Urgent call=140

Charges for Lightning call=60
```

# 3) Package and Interface:

**i. Write a program to make a package Balance in which has account class with display balance method in it. Import balance package in another program to access Display balance method of account class.**

**PROGRAM:**

```
/* In the file Account.java*/
package Balance;
public class Account
{
        double p, i, r, balance;
        int t;
        public Account (double pr, int ti, double ra)
        {
                p=pr;
                t=ti;
                r=ra;
        }
        public void cal ()
        {
                balance=p*r*t;
        }
        public void Disply_Balance ()
        {
                System.out.println("\n\nPrincipal Amount: "+p+"Rs\nTime:
                "+t+"Years\n\nCurrent Balance: "+balance+"Rs");
        }
}
```

```
/*In the file Pack.java*/
import Balance. *;
public class Pack
{
        public static void main (String args [])
        {
                Account b1=new Account (5000,2,0.12);
                b1.cal ();
                b1. Disply_Balance ();
        }
}
```

## STEPS FOR EXECUTION:

1. Go to your workspace, Right-click->Open terminal here
2. Create a directory named Balance (mkdir Balance)
3. Go to Balance directory (cd Balance)
4. Create a file named Account.java (vi Account.java)
5. Type the program given above and save it (esc+Shift+: wq!)
6. Compile the program using the command javac Account.java
7. Now go back to your workspace (cd ..)
8. Create a file named Pack.java (vi Pack.java)
9. Type the program given above and save it (esc+Shift+: wq!)
10. Compile the program using the command javac Pack.java
11. Run the program using the command java Pack

## OUTPUT:

```
Principal Amount: 5000.0Rs
Time: 2Years
Current Balance: 1200.0Rs
```

## ii. Create the dynamic stack by implementing the interfaces that defines Push() and Pop() methods.

## PROGRAM:

```
interface instack
{
        void push (int item);
        int pop ();
}
class dstack implements instack
{
        private int stk [];
        private int tos;

        dstack (int size)
        {
                stk=new int[size];
                tos=-1;
        }

        public void push (int item)
        {       if (tos==stk. length-1)
                {
                        int temp []=new int[stk.length*2];
                        for (int i=0; i<stk. length; i++)
                                temp[i]=stk[i];
                        stk=temp;
                        stk[++tos]=item;
                }
                else
                {
                        stk[++tos]=item;
                }
        }

        public int pop ()
        {        if(tos<0)
                {
                        System.out.println("stack underflow");
                        return 0;
                }
                else
                {
                        return stk[tos--];
                }
        }
}
```

```
public class Dyn_stack
{
	public static void main (String args [])
	{
		dstack mystack1=new dstack (5);
		dstack mystack2=new dstack (8);
		for (int i=0;i<20;i++)
		mystack1.push(i);
		for (int i=0;i<20;i++)
		mystack2.push(i);
		System.out.print("\t Elements in stack1 -> ");
		for (int i=0;i<20;i++)
		System.out.print(mystack1.pop () +" ");
		System.out.println();
		System.out.print("\t Elements in stack2 -> ");
		for (int i=0;i<20;i++)
		System.out.print(mystack2.pop () +" ");
		System.out.println();
	}
}
```

## STEPS FOR EXECUTION:

1. Go to your workspace, Right-click->Open terminal here
2. Create a file named Dyn_stack.java (vi Dyn_stack.java)
3. Type the program given above and save it (esc+Shift+: wq!)
4. Compile the program using the command javac Dyn_stack.java
5. Run the program using the command java Dyn_stack

## OUTPUT:

```
Elements in stack1 -> 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Elements in stack2 -> 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
```

# 4.Exception handling

**On a single track two vehicles are running. As vehicles are going on same direction there is no problem. If the vehicles are running in different direction, there is a chance of collision. To avoid collision, write a java program using Exception handling.**

## PROGRAM:

```
class collision
{
	String i, j;

	collision (String a, String b)
	{
		i=a;
		j=b;
	}

	void check ()
	{
		try
		{
			if(i==j)
			{
				System.out.println("The two vehicles are moving in same
			direction, hence no problem");
			}
			else
			{
				throw new Exception ("The two vehicles are moving in
			different directions, so collision occurs");
			}
		} catch (Exception e)
		{
			System.out.println(e);
		}
	}
}

public class exception
{
	public static void main (String args [])
	{
		collision s=new collision ("north”, “north");
		collision n=new collision ("north”, “east");
		s. check ();
		System.out.println();
```

```
            n. check ();
        System.out.println();
        }
}
```

## STEPS FOR EXECUTION:

1. Go to your workspace, Right-click->Open terminal here
2. Create a file named exception.java (vi exception.java)
3. Type the program given above and save it (esc+Shift+: wq!)
4. Compile the program using the command javac exception.java
5. Run the program using the command java exception

## OUTPUT:

The two vehicles are moving in same direction, hence no problem
java.lang.Exception: The two vehicles are moving in different

# 5.MULTITHREADING

## 1. Write a Java program to create five threads with different priorities. Send two threads of highest priority to sleep state. Check the aliveness of the threads and mark which thread is long lasting

## PROGRAM:

```
class NewThread
{
	public static void main(String args[])
	{	Thread.currentThread().setPriority(Thread.MAX_PRIORITY);
		MulThread m1=new MulThread("one",Thread.NORM_PRIORITY-1);
		MulThread m2=new MulThread("two",Thread.MAX_PRIORITY);
		MulThread m3=new MulThread("three",Thread.NORM_PRIORITY+2);
		MulThread m4=new MulThread("four",Thread.NORM_PRIORITY+4);
		MulThread m5=new MulThread("five",Thread.MIN_PRIORITY+1);

		try
		{
			Thread.sleep(500);
		}
		catch(InterruptedException e)
		{
			System.out.println("main thread interrupted");
		}
	System.out.println("Thread one is:"+m1.t.isAlive());
	System.out.println("Thread two is:"+m2.t.isAlive());
	System.out.println("Thread three is:"+m3.t.isAlive());
	System.out.println("Thread four is:"+m4.t.isAlive());
	System.out.println("Thread five is:"+m5.t.isAlive());
```

```
try
{
        System.out.println("waiting for thread to finish");
        m1.t.join();
        m2.t.join();
        m3.t.join();
        m4.t.join();
        m5.t.join();
}
catch(InterruptedException e)
{
        System.out.println("main thread interrupted");
}

System.out.println("thread one is:"+m1.t.isAlive());
System.out.println("thread two is:"+m2.t.isAlive());
System.out.println("thread three is:"+m3.t.isAlive());
System.out.println("thread four is:"+m4.t.isAlive());
System.out.println("thread five is:"+m5.t.isAlive());
System.out.println();
System.out.println("priority of one:"+m1.t.getPriority());
System.out.println("priority of two:"+m2.t.getPriority());
System.out.println("priority of three:"+m3.t.getPriority());
System.out.println("priority of four:"+m4.t.getPriority());
System.out.println("priority of five:"+m5.t.getPriority());
System.out.println();
System.out.println(MulThread.last+" is long lasting thread");
```

```
        }
}


class MulThread implements Runnable
{
        static String last;
        String name;
        Thread t;
        MulThread(String n,int p)
        {
                name=n;
                t=new Thread(this, name);
                t.setPriority(p);
                System.out.println(name+" started");
                System.out.println("new thread: "+t);
                t.start();
        }

public void run()
 {
        try
        {
                if((t.getPriority()==9)||(t.getPriority()==10))
                {
                        Thread.sleep(1000);
                        System.out.println(t.getName()+" is going to sleep");
                }
```

```
                for(int i=0;i<5;i++)
                {
                        System.out.println(name+":"+i);
                        Thread.sleep(500);
                }
  }

        catch(InterruptedException e)
        {
                System.out.println(name+" thread interrupted");

        }
        last=name;
        System.out.println(name+" exiting");

 }
}
```

## STEPS FOR EXECUTION:

1. Go to your workspace, Right-click->Open terminal here
2. Create a file named NewThread.java (vi NewThread.java)
3. Type the program given above and save it (esc+Shift+: wq!)
4. Compile the program using the command javac NewThread.java
5. Run the program using the command java NewThread

## 2. Write a multi-threaded Java program to implement producer consumer problem

## PROGRAM:

```
class Q
{
  int n;
    boolean valueSet=false;

synchronized int get()
  {
    while(!valueSet)

   try
    {
    Thread.sleep(1000);
    wait();
    }
      catch(InterruptedException e)
      {
           System.out.println("Interpted Exection Caught");
      }

    System.out.println("Got:"+n);
    valueSet=false;
    notify();
    return n;
  }
```

```
synchronized void put(int n)
  {
    while(valueSet)

      try
      {
            Thread.sleep(1000);
            wait();
      }
    catch(InterruptedException e)
      {
            System.out.println("Interpted Exection Caught");
      }

  this.n=n;
valueSet=true;
System.out.println("Put :"+n);
notify();

 }
}

class prod implements Runnable
{
  Q q;
prod(){}
  prod(Q q)
  {
```

```
    this.q=q;
    new Thread(this,"Producer").start();
  }

public void run()
{
  int i=0;
while(true)
  {
    q.put(i++);
  }
}
}

class cons implements Runnable
{
  Q q;
cons(){}
  cons(Q q)
  {
    this.q=q;
    new Thread(this,"Consumer").start();
  }

public void run()
{

while(true)
```

```
  {
    q.get();
  }
}
}

Public class pcfix
{
  public static void main(String args[])
    {
      Q q=new Q();
    new prod(q);
    new cons(q);
      System.out.println("Press Ctrl+C to Stop");
}
}
```

## STEPS FOR EXECUTION:

1. Go to your workspace, Right-click->Open terminal here
2. Create a file named pcfix.java (vi pcfix.java)
3. Type the program given above and save it (esc+Shift+: wq!)
4. Compile the program using the command javac pcfix.java
5. Run the program using the command java pcfix

# 6. Applets and Event Handling

## 1. Design an applet which uses Card Layout with 3 Buttons When the user clicks on any button the background layout color must change

**PROGRAM :**

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

/*<applet code="CardLayoutExm" width=300 height=400>
</applet>
*/

public class CardLayoutExm extends Applet implements ActionListener
{
 JPanel cardPanel;
 JPanel firstp,secondp,thirdp,fourthp;
 JPanel buttonp;
 JButton first,second,third;
 CardLayout ourLayout;

 public void init()
 {
  cardPanel=new JPanel();
  ourLayout=new CardLayout();
  cardPanel.setLayout(ourLayout);
```

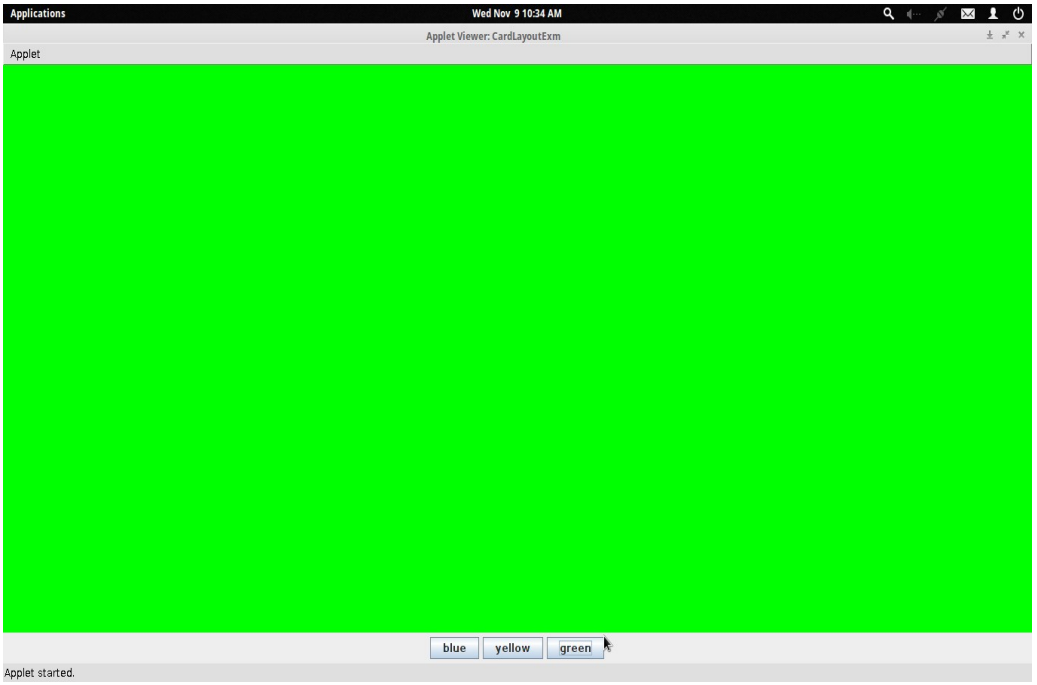
```
firstp=new JPanel();
firstp.setBackground(Color.blue);

secondp=new JPanel();
secondp.setBackground(Color.yellow);

thirdp=new JPanel();
thirdp.setBackground(Color.green);

fourthp=new JPanel();

first=new JButton("blue");
first.addActionListener(this);

second=new JButton("yellow");
second.addActionListener(this);

third=new JButton("green");
third.addActionListener(this);

buttonp=new JPanel();
buttonp.add(first);
buttonp.add(second);
buttonp.add(third);

this.setLayout(new BorderLayout());
this.add(buttonp,BorderLayout.SOUTH);
this.add(cardPanel,BorderLayout.CENTER);
```

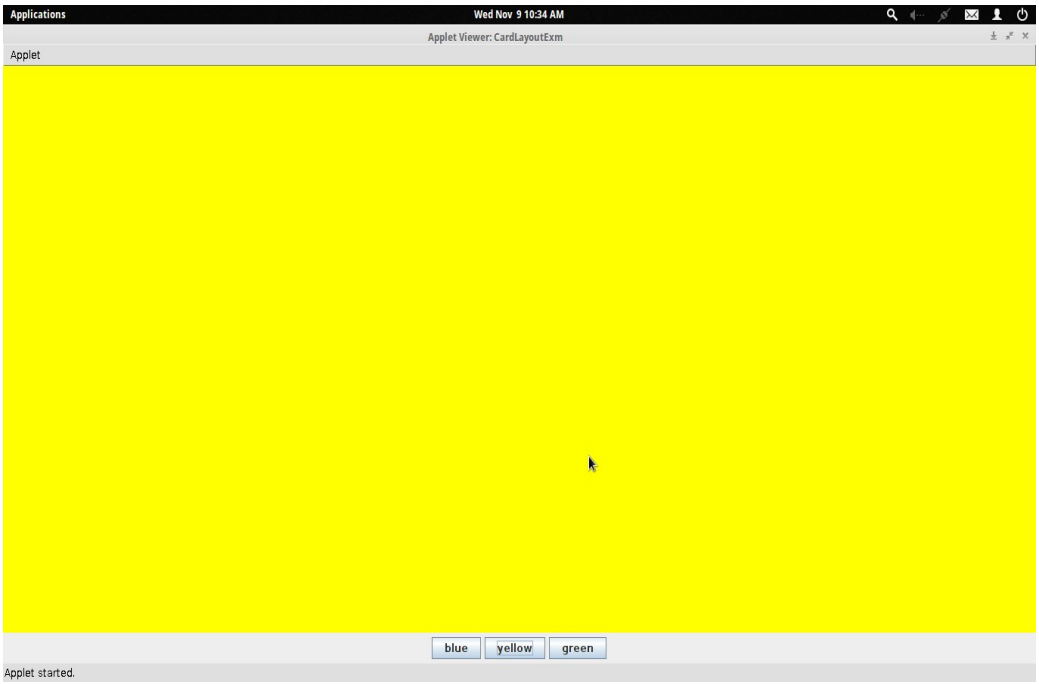
```
 cardPanel.add(fourthp,"Fourth");
 cardPanel.add(firstp,"First");
 cardPanel.add(secondp,"Second");
 cardPanel.add(thirdp,"Third");
}

public void actionPerformed(ActionEvent e)
{
 if(e.getSource()==first)
  ourLayout.show(cardPanel,"First");

 if(e.getSource()==second)
  ourLayout.show(cardPanel,"Second");

 if(e.getSource()==third)
  ourLayout.show(cardPanel,"Third");
 }

}
```

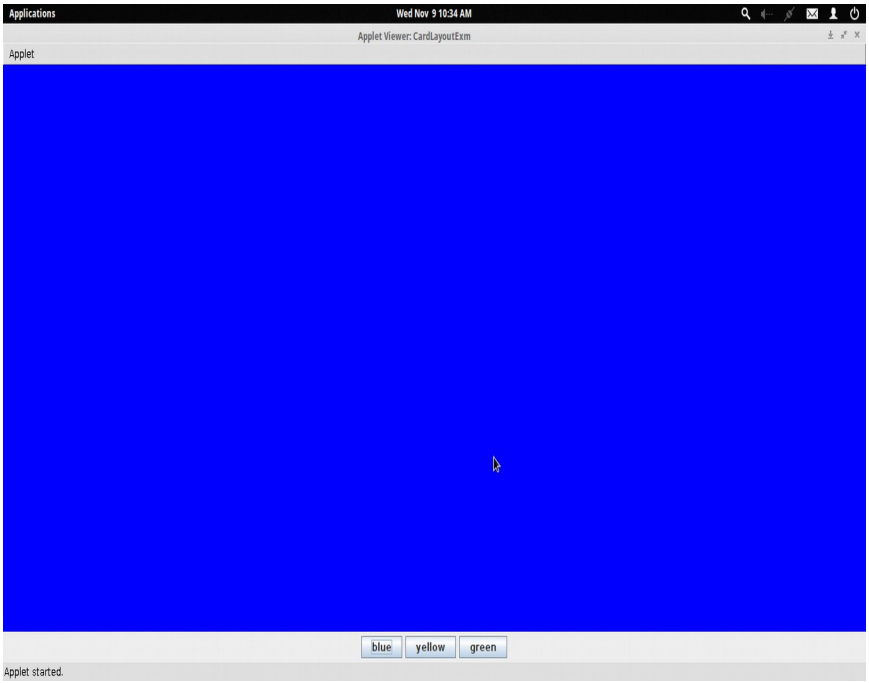
## STEPS FOR EXECUTION:

1. Go to your workspace, Right-click->Open terminal here
2. Create a file named CardLayoutExm.java (vi CardLayoutExm.java)
3. Type the program given above and save it (esc+Shift+: wq!)
4. Compile the program using the command  javac CardLayoutExm.java
5. Run the program using the command appletviewer CardLayoutExm.html

# OUTPUT:


Applications
Wed Nov 9 10:34 AM
Applet Viewer: CardLayoutExm
Applet
blue
yellow
green
Applet started.


Applications
Wed Nov 9 10:34 AM
Applet Viewer: CardLayoutExm
Applet
blue
yellow
green
Applet started.


Applications
Wed Nov 9 10:34 AM
Applet Viewer: CardLayoutExm
Applet
blue
yellow
green
Applet started.

## 2. Create an applet to handle all mouse events

**PROGRAM :**

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

public class MouseEvents extends Applet implements MouseListener,MouseMotionListener
{

String msg="";
int mouseX=0,mouseY=0;

public void init()
{
addMouseListener(this);
addMouseMotionListener(this);
}

public void mouseClicked(MouseEvent me)
{
mouseX=0;
mouseY=10;
msg="Mouse Clicked";
repaint();
}

public void mouseEntered(MouseEvent me)
```

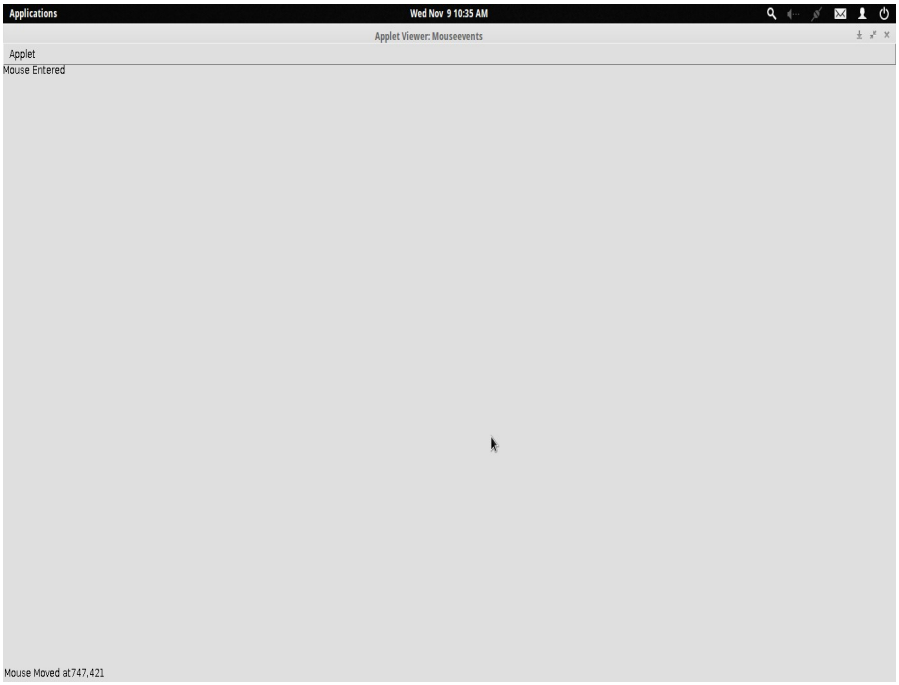
```
{
mouseX=0;
mouseY=10;
msg="Mouse Entered";
repaint();
}

public void mouseExited(MouseEvent me)
{
mouseX=0;
mouseY=10;
msg="Mouse exited";
repaint();
}

public void mousePressed(MouseEvent me)
{
mouseX=me.getX();
mouseY=me.getY();
msg="Down";
repaint();
}

public void mouseReleased(MouseEvent me)
{
mouseX=me.getX();
mouseY=me.getY();
msg="Up";
```

```
repaint();
}

public void mouseDragged(MouseEvent me)
{
mouseX=me.getX();
mouseY=me.getY();
msg="*";
repaint();
showStatus("Mouse Dragged at "+mouseX+","+mouseY);
}

public void mouseMoved(MouseEvent me)
{
showStatus("Mouse Moved at "+me.getX()+","+me.getY());
}

public void paint(Graphics g)
{
g.drawString(msg,mouseX,mouseY);
}

}
```

## STEPS FOR EXECUTION:

1. Go to your workspace, Right-click->Open terminal here
2. Create a file named MouseEvents.java (vi MouseEvents.java)
3. Type the program given above and save it (esc+Shift+: wq!)
4. Compile the program using the command javac MouseEvents.java
5. Run the program using the command appletviewer MouseEvents.html

# OUTPUT:


Applications
Wed Nov 9 10:35 AM
Applet Viewer: Mouseevents
Applet
Mouse Entered
Mouse Moved at747,421

# 7. SERVLETS

## 1. Program to accept username, address and display them in a web page by passing parameters

**PROGRAM :**

```
import javax.servlet.*;

import javax.servlet.http.*;

import java.io.*;

public class WebForm extends HttpServlet{

   public void doGet(HttpServletRequest request,HttpServletResponse response) throws
IOException,ServletException{

                String name, addr;


                response.setContentType("text/html");

                name=request.getParameter("uname");

                addr=request.getParameter("address");

                PrintWriter out=response.getWriter();

                out.println("<html><body-bgcolor='#ffffff' text='#000000'>");

                out.println("<h1  align=center> Welcome "+name+
"</h1><hr>Address:"+addr);

                out.close();

        }

}
```

**HTML FILE:**

```
<html>
<head>
<title>Greeting...</title>
</head>
<body-bgcolor='#ffffff' text='#000000'>
<h1 align=center>GREETING A USER</h1>
<hr>
<form method=get action="http://localhost:8080/examples/servlets/servlet/WebForm">



<table>
<tr>
        <td align="right"><b>NAME:</b></td>
        <td><input type=text name=uname /></td>
</tr>
<tr>
        <td><b>ADDRESS:</b></td>
        <td><input type=text name=address /></td>
</tr>


</table>
<input type=submit value=SUBMIT />
<input type=reset value=CLEAR />
<hr>
</form>
</body>
```

</html>

## STEPS FOR EXECUTION:

- **Compile the program using the command**
  javac WebForm.java -classpath /usr/share/java/servlet-api.jar

- **copy this to the specified path**
  sudo cp WebForm.class /usr/share/tomcat7-examples/examples/WEB-INF/classes/

- **Create lab.html and store**
  sudo cp WebForm.html /var/www

- **edit web.xml (make a backup web.xml)**
  cd /usr/share/tomcat7-examples/examples/WEB-INF
  sudo gedit web.xml

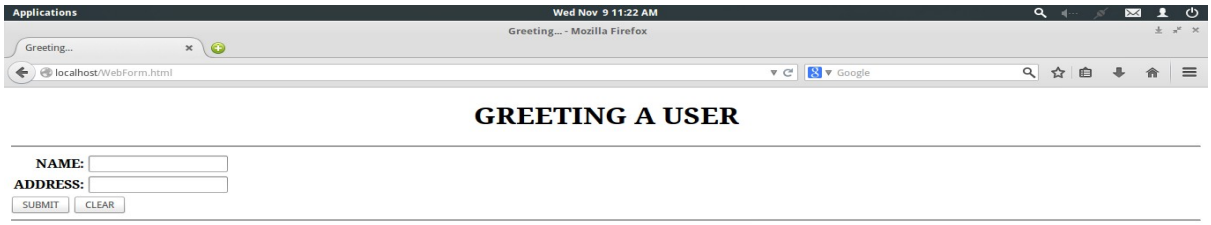
- create a new <servlet> entry

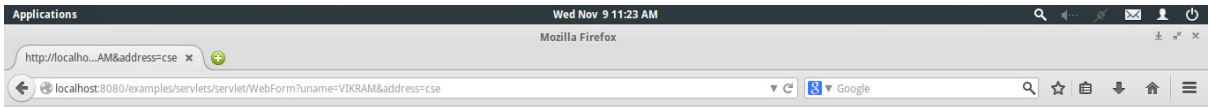
- create a new <servlet-mapping> entry

- **Open the following link in your browser**

  localhost/WebForm.html

## OUTPUT:






# Welcome VIKRAM

---

Address:cse

## 2. Program to request server information viz Request Method URL, Protocol and Remote address

## PROGRAM :

```
import javax.servlet.*;
import java.io.*;
import javax.servlet.http.*;
public class lab2 extends HttpServlet
{
public void doGet(HttpServletRequest  request, HttpServletResponse response)throws IOException,ServletException
{
response.setContentType("text/html");
PrintWriter out=response.getWriter();

out.println("<html><head>");
out.println("<title>Server Information</title>");
out.println("</head>");
out.println("<body bgcolor='#ffffff',text='#000000'>");
out.println("<h1 align=center> SERVER INFORMATTION");
out.println("<hr><br><center><table border=5><tr>");

out.println("<td><b>Request Method</b></td>");
out.println("<td>");
out.println(request.getMethod());
out.println("</td></tr>");
out.println("<tr>");
```

```
out.println("<td><b> URL</b></td>");
out.println("<td>");
out.println(request.getRequestURL());
out.println("</td></tr>");
out.println("<tr>");

out.println("<td><b>Protocol </b></td>");
out.println("<td>");
out.println(request.getProtocol());
out.println("</td></tr>");
out.println("<tr>");

out.println("<td><b>Remote Address<b></td>");
out.println("<td>");
out.println(request.getRemoteAddr());
out.println("</td></tr></table>");
out.println("<br><hr>");
out.println("</body></html>");
}
}
```

## STEPS FOR EXECUTION :

- **Compile the program using the command**
  javac lab2.java -classpath /usr/share/java/servlet-api.jar
- **copy this to the specified path**
  sudo cp lab2.class /usr/share/tomcat7-examples/examples/WEB-INF/classes/
- **edit web.xml (make a backup web.xml)**

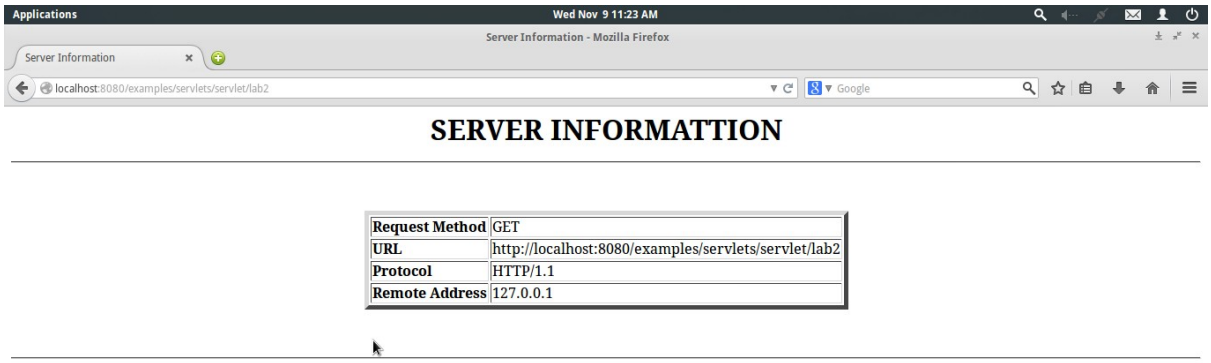
cd /usr/share/tomcat7-examples/examples/WEB-INF
sudo gedit web.xml

- create a new <servlet> entry

- create a new <servlet-mapping> entry

- **Open the following link localhost:8080**
- **click on examples**
- **servlet examples**
- **execute any one**
- **change the URL to lab2**

# OUTPUT:



# 8. SWINGS AND JDBC

**Write a Java program to implement Client Server interaction (Client requests a file, Server responds to client with contents of that file which is then displayed on the screen by Client) –Socket programming**

**PROGRAM :**

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class login extends JFrame implements ActionListener
{
        JButton SUMBIT;
        JPanel panel;
        JLabel label1,label2;
        final JTextField text1,text2;
        login()
        {
                label1=new JLabel();
                label1.setText("username");
                text1=new JTextField(15);
                 label2=new JLabel();
                label2.setText("password");
                 text2=new JTextField(15);

                SUMBIT=new JButton("SUMBIT");
                panel=new JPanel(new GridLayout(3,1));
                panel.add(label1);
                panel.add(text1);
                panel.add(label2);
```

```
                panel.add(text2);
               panel.add(SUMBIT);
                add(panel,BorderLayout.CENTER);
                SUMBIT.addActionListener(this);
                setTitle("login form");
        }
public void actionPerformed(ActionEvent ae)
{
                String value1=text1.getText();
                String value2=text2.getText();
        java.sql.Connection conn=null;
        try
                {
Class.forName("com.mysql.jdbc.Driver").newInstance();
conn=java.sql.DriverManager.getConnection("jdbc:mysql://localhost/sit?
user=root&password=");
         }
                catch(ClassNotFoundException e)
                {
                System.out.println("errorindriverloader"+e);
        System.exit(1);
                }
        catch(Exception e)
        {
                System.out.println("error inconnction"+e);
        System.exit(0);
                }
        System.out.println("connection established");
         try
```

```
        {
                java.sql.Statement s=conn.createStatement();
                String query="select * from table1 where uname='"+value1+"'
        and password='"+value2+"'";
java.sql.ResultSet r=s.executeQuery(query);
         r.next();
              int x=r.getRow();
              if (x>0)
             {
        JOptionPane.showMessageDialog(null,"HELLOOOOO");                    }
                 else
                 {
        JOptionPane.showMessageDialog(this,"incoreect login of
        password","error",JOptionPane.ERROR_MESSAGE);                       }

        }  catch(Exception e)
               {
                 System.out.println(e);
                        System.exit(0);                                          }
        }
class Demo
{
        public static void main(String args[])
        {
                try
                {
                        login frame=new login();
                        frame.setSize(300,100);
                        frame.setVisible(true);
                }
```

```
                catch(Exception e)
                {
JOptionPane.showMessageDialog(null,e.getMessage());
}
}
}
```

## STEPS FOR EXECUTION:

1. Go to your workspace, Right-click->Open terminal here
2. Create a file named Demo.java (vi Demo.java)
3. Type the program given above and save it (esc+Shift+: wq!)
4. Compile the program using the command Javac Demo.java
5. **Type the following commands**
6. mysql –u root –p
7. Enter Password $: root123
8. **Create database using :** Create database sit;
   use sit;

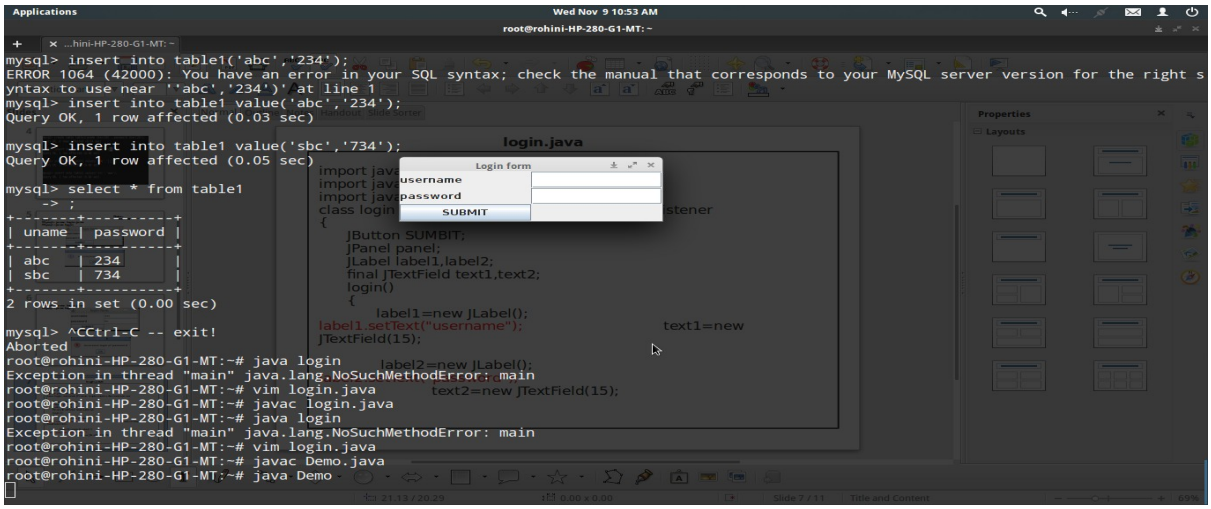
9. **Create table using**: create table table1(uname char (30), password char (20);

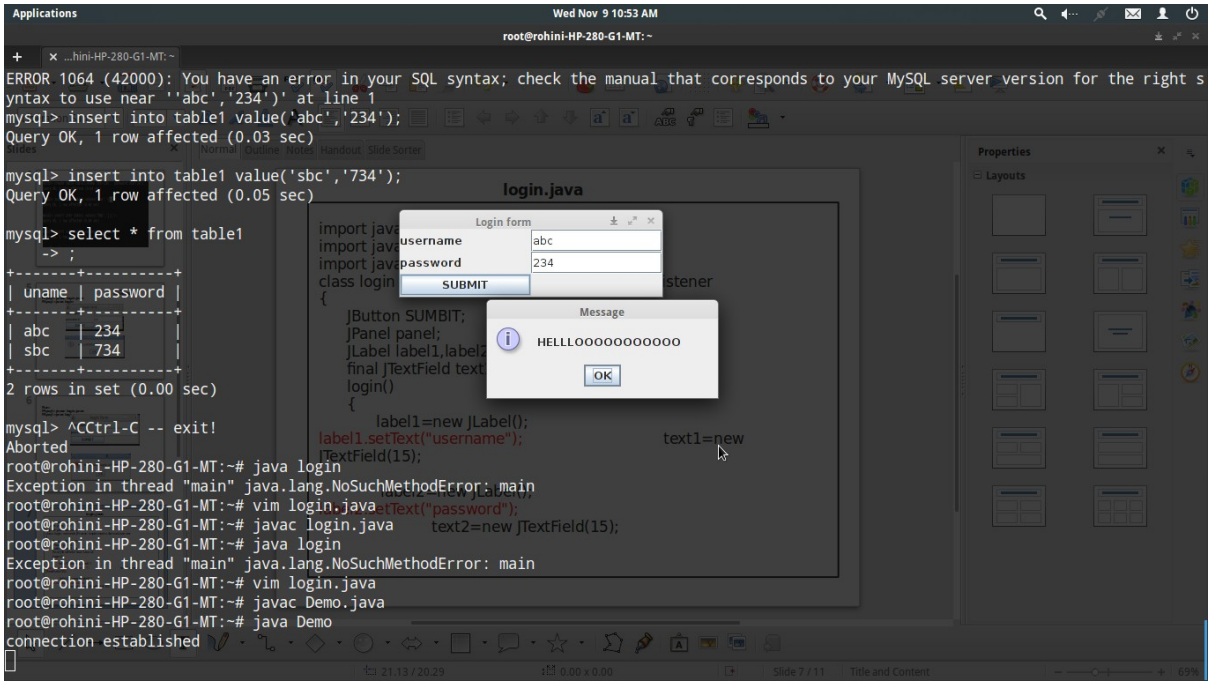
**10. Insert data into table using**: insert into table1 values('abc','pass1'); repeat this step to enter several data

**11. To view table contents**: select * from table1

**12. Run the program using the command** :java Demo

## OUTPUT:


Applications
Wed Nov 9 10:53 AM
root@rohini-HP-280-G1-MT: ~
mysql> insert into table1('abc','234');
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right s
yntax to use near ''abc','234')' at line 1
mysql> insert into table1 value('abc','234');
Query OK, 1 row affected (0.03 sec)
mysql> insert into table1 value('sbc','734');
Query OK, 1 row affected (0.05 sec)
mysql> select * from table1
-> ;
+-------+----------+
| uname | password |
+-------+----------+
| abc | 234 |
| sbc | 734 |
+-------+----------+
2 rows in set (0.00 sec)
mysql> ^CCtrl-C -- exit!
Aborted
root@rohini-HP-280-G1-MT:~# java login
Exception in thread "main" java.lang.NoSuchMethodError: main
root@rohini-HP-280-G1-MT:~# vim login.java
root@rohini-HP-280-G1-MT:~# javac login.java
root@rohini-HP-280-G1-MT:~# java login
Exception in thread "main" java.lang.NoSuchMethodError: main
root@rohini-HP-280-G1-MT:~# vim login.java
root@rohini-HP-280-G1-MT:~# javac Demo.java
root@rohini-HP-280-G1-MT:~# java Demo
login.java
Login form
username
password
SUBMIT
Properties
Layouts


Applications
Wed Nov 9 10:53 AM
root@rohini-HP-280-G1-MT: ~
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right s
yntax to use near ''abc','234')' at line 1
mysql> insert into table1 value('abc','234');
Query OK, 1 row affected (0.03 sec)
mysql> insert into table1 value('sbc','734');
Query OK, 1 row affected (0.05 sec)
mysql> select * from table1
-> ;
+-------+----------+
| uname | password |
+-------+----------+
| abc | 234 |
| sbc | 734 |
+-------+----------+
2 rows in set (0.00 sec)
mysql> ^CCtrl-C -- exit!
Aborted
root@rohini-HP-280-G1-MT:~# java login
Exception in thread "main" java.lang.NoSuchMethodError: main
root@rohini-HP-280-G1-MT:~# vim login.java
root@rohini-HP-280-G1-MT:~# javac login.java
root@rohini-HP-280-G1-MT:~# java login
Exception in thread "main" java.lang.NoSuchMethodError: main
root@rohini-HP-280-G1-MT:~# vim login.java
root@rohini-HP-280-G1-MT:~# javac Demo.java
root@rohini-HP-280-G1-MT:~# java Demo
connection established
login.java
Login form
username
abc
password
234
SUBMIT
Message
HELLLOOOOOOOOOO
OK
Properties
Layouts

# 9.NETWORKING

## Write a Java Program to implement Client Server interaction.

## Program:

## DailyAdviceServer.java:

```
import java.io.*;
import java.net.*;
public class DailyAdviceServer
{
   String[] adviceList = {"OS", "DBMS", "Java", "CN", "ADA"};

   public void go() {
      try {
         ServerSocket serverSock = new ServerSocket(4242);
         while (true)
         {
            Socket sock = serverSock.accept();

            PrintWriter writer = new PrintWriter(sock.getOutputStream());
            String advice = getAdvice();
writer.println(advice);
            writer.close();
            System.out.println(advice);
         }
      } catch (IOException ex)
      {
         ex.printStackTrace();
      }
```

```
    }

    private String getAdvice() {
        int random = (int) (Math.random() * adviceList.length);
        return adviceList[random];
    }

    public static void main(String[] args)
    {
        DailyAdviceServer server = new DailyAdviceServer();
        server.go();
    } }
```

## DailyAdviceClient.java:

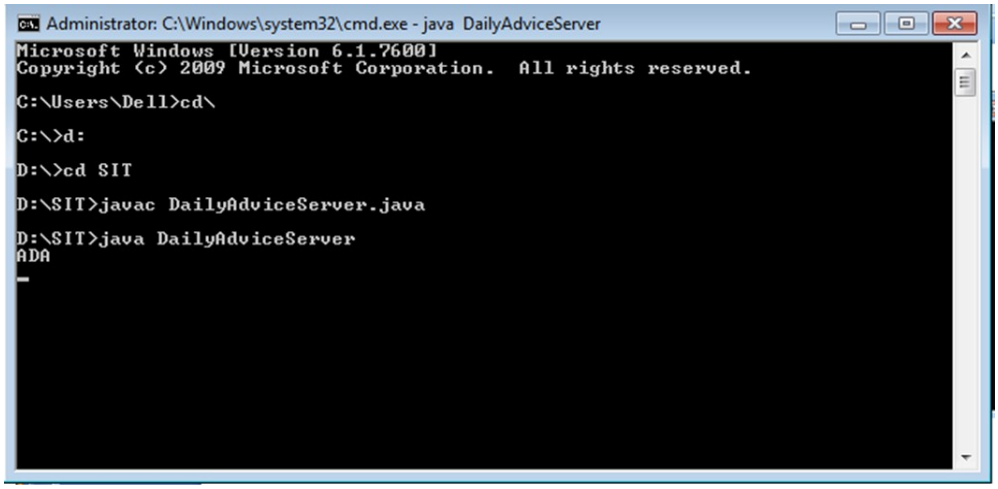
```
import java.io.*;
import java.net.*;
public class DailyAdviceClient
{
    public void go() {
        try {
            Socket s = new Socket("127.0.0.1", 4242);
            InputStreamReader streamReader = new InputStreamReader(s.getInputStream());
            BufferedReader reader = new BufferedReader(streamReader);

            String advice = reader.readLine();
```

```
            System.out.println("Subject is: " + advice);
            reader.close();
        }
catch (IOException ex)
        {
            ex.printStackTrace();
        }
    }

    public static void main(String[] args)
    {
        DailyAdviceClient client = new DailyAdviceClient();
        client.go();
    }
}
```

## STEPS FOR EXECUTION AND OUTPUT:

1. Go to your workspace, Right-click->Open terminal here
2. Create a file named DailyAdviveServer.java (vi DailyAdviceServer.java)
3. Type the program given above and save it (esc+Shift+: wq!)
4. Compile the program using the command  javac DailyAdviceServer.java
5. Repeat the above steps in a new terminal to create and compile DailyAdviceClient.java
6. Run the Server program using the command java DailyadviceServer
7. Now run the Client program using the comman java DailyAdviceClient

## DailyAdviceServer:




## DailyAdviceClient: