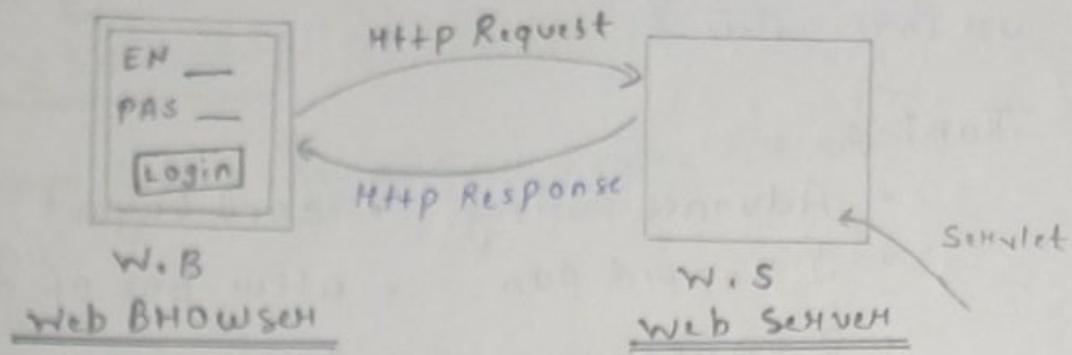
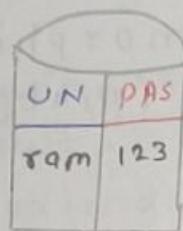


Syllabus

HTML
CSS
JS



- HTML (Hyper Text Markup Language) - used for making a form.
- CSS (cascading style sheets) - for looks attractive
- JS (Java Script) - for validation.



JDBC (Java database connectivity)

Database

Servlet advance version → JSP (Java Server Page)

* Session
* Cookies

- AT (Apache Tomcat) - इसमें notepad में project कूंसे बनेगा वो देखेंगे।
- NB (NetBeans) - इसमें भी project बनेगा
NB - JSP में netBeans की help से project बनेगा
- session cookie की help से कूंसे बनेगा

MVC (Model View Controller) Architecture :- स्ट्री

उसके एक same mannah में coding करने के लिए
एक फ्रेम आता है

Topics

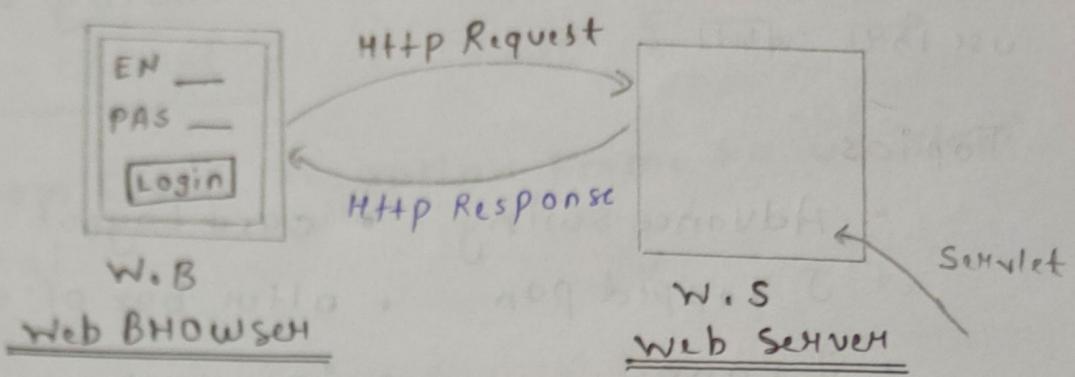
- Advance swing • card Layout
- J Tapped pan • alter box of different types
- J Panel • inner class
- J Table • Adapter class
- Lambda Expression
- Big integer
- Whapper class
- Templet class

Object class - Advance Polymorphism, instance of keyword , GC (garbage collector), method channing.

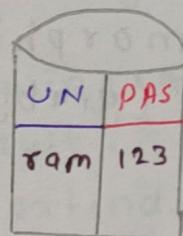
*NOTE Boolean - Whapper class - यह W.C हो
1st letter capital
hota hai

Syllabus

HTML
CSS
JS



- HTML (Hyper Text Markup Language) - used for making a form.
- CSS (cascading style sheets) - for looks attractive
- JS (Java Script) - for validation.



JDBC (Java database connectivity)

Database

Servlet advance version → JSP (Java Server Page)

* Session
* cookies

- AT (Apache Tomcat) - इसमें notepad में project कूसे लेगा
- NB (NetBeans) - इसमें शी project बनेगा
NB - JSP में NetBeans की help से project करेगा
- session cookie की help से कूसे लेगा

MVC (Model view controller) Architecture :- समीक्षा

User को एक same mannah में coding करने के लिए
use किया जाता है

Topics

- Advance swing
- card Layout
- J Tapped pan
- altor box of different types
- J Pannel
- inner class
- J Table
- Adapter class
- Lambda Expression
- Big integer
- Whapper class
- Templete class

Object class - Advance Polymorphism, instance of keyword , GC (garbage collector), method channing.

* NOTE Boolean - Whapper class - समीक्षा w.c के
1st letter capital hota hai

26/07/23

Wednesday

Advance swing concept

- * Kabhi bhi ham frame ke andar frame ka use Nahi kar sakte hai.
- * Yadi esa karte hai to? illegal argument exception message aakar display ho jaye gi.
- * Frame ke andar panel ka use kar sakte hai.
- * Panel ke andar panel ko add kar sakte hai.
- * Panel ke andar frame ko add nahi kar sakte hai.
- * Panel me default layout FlowLayout hota hai.

①

```
import java.swing.*;
import java.awt.*;
class FDemo extends JFrame
{
    JButton b1, b2, b3, b4, b5, b6;
    FDemo()
    {
        GridLayout gr=new GridLayout(2,3);
        setLayout(gr);
        b1=new JButton("Button 1");
        add(b1);
        b2=new JButton("Button 2");
        add(b2);
        b3=new JButton("Button 3");
        add(b3);
```

```

b4 = new JButton("Button 4");
add(b4);

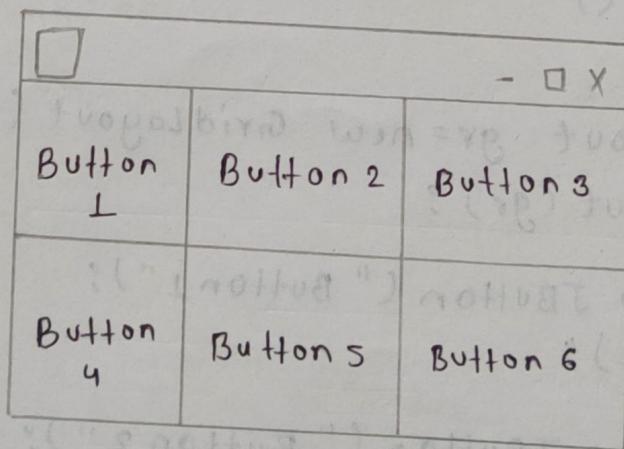
b5 = new JButton("Button 5");
add(b5);

b6 = new JButton("Button 6");
add(b6);
}

class Demo
{
public static void main(String args[])
{
    FDemo f=new FDemo();
    f.setVisible(true);
    f.setSize(500,500);
    f.setLocation(100,100);

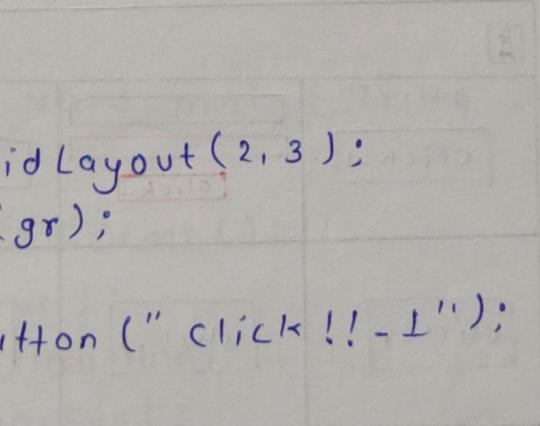
    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
}

```



yadi ham esa chate hain ki 2nd Button 3 place par
ek form aana chahiye to

② import javax.swing.*;
 class JPDemo extends JPanel → अदि भट्टा JPanel की जगह
 {
 JTextField tx1, tx2;
 JButton b1;
 JPDemo()
 {
 tx1 = new JTextField(10);
 add(tx1);
 tx2 = new JTextField(10);
 add(tx2);
 b1 = new JButton("click!!...");
 add(b1);
 }
 }
 3

import javax.swing.*;
 import java.awt.*;
 class FDemo extends JFrame
 {
 JButton b1, b2, b3, b4, b5, b6;
 JPDemo jp;
 GridLayout gr;
 FDemo()
 {
 gr = new GridLayout(2, 3);
 setLayout(gr);
 b1 = new JButton("click!!-1");
 add(b1);
 }
 }


jp = new JPDemo();
add(jp);

मर्दां पर JButton bhi ek class है अतः
JPDemo भी एक class है तो इस b
में जगह jp का use कर सकते हैं

b3 = new JButton("click!!-3");
add(b3);

b4 = new JButton("click!!-4");
add(b4);

b5 = new JButton("click!!-5");
add(b5);

b6 = new JButton("click!!-6");
add(b6);

3

class Demo
{

public static void main(String args[])

{

FDemo f = new FDemo();

f.setVisible(true);

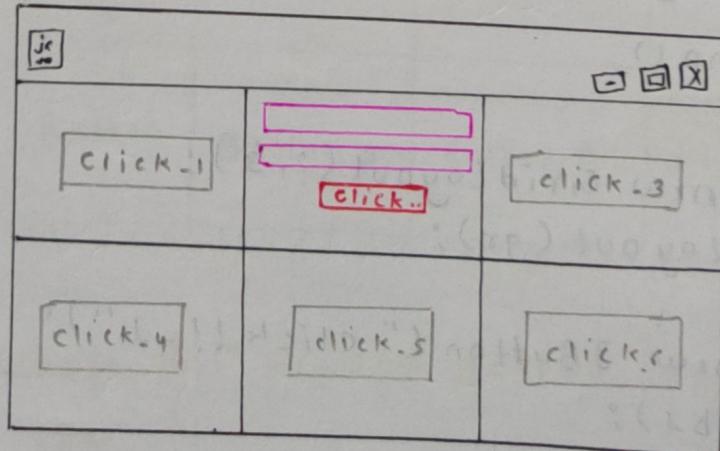
f.setSize(500, 500);

f.setLocation(100, 100);

f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

3

3



27/07/23

thursday * यदि हम ऐसा चाहते हैं कि Button पर click करने पर background color change हो तो

①

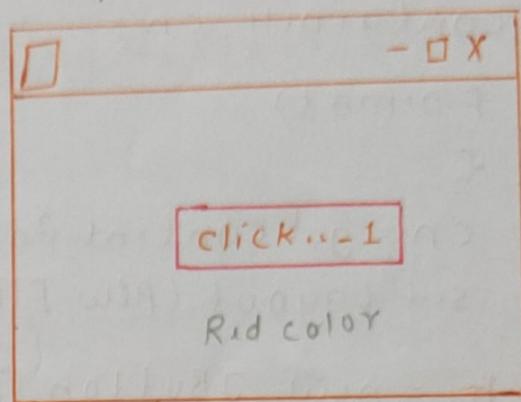
```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

class FDemo extends JFrame implements ActionListener
{
    JButton b1;
    Container cn;

    FDemo()
    {
        cn = getContentPane();
        setLayout(new FlowLayout());
        b1 = new JButton("click!!-1");
        add(b1);
        b1.addActionListener(this);
    }

    public void actionPerformed(ActionEvent e)
    {
        cn.setBackground(Color.red);
    }
}

class Demo
{
    public static void main(String args)
    {
        FDemo f = new FDemo();
        f.setVisible(true);
        f.setLocation(100, 100);
        f.setSize(500, 500);
        f.setDefaultCloseOperation(f.EXIT_ON_CLOSE);
    }
}
```



②

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

```
class FDemo extends JFrame implements ActionListener
```

```
{ JButton b1, b2, b3;
```

```
Container cn;
```

```
FDemo()
```

```
{
```

```
cn = getContentPane();
```

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

```
b1 = new JButton("Button-1");
```

```
add(b1);
```

```
b2 = new JButton("Button-2");
```

```
add(b2);
```

```
b3 = new JButton("Button-3");
```

```
add(b3);
```

```
b1.addActionListener(this);
```

```
b2.addActionListener(this);
```

```
b3.addActionListener(this);
```

```
}
```

```
public void actionPerformed(ActionEvent e)
```

```
{
```

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

```
{
```

```
cn.setBackground(Color.red);
```

```
}
```

```
if(e.getSource() == b2)
```

```
{
```

```
cn.setBackground(Color.green);
```

```
}
```

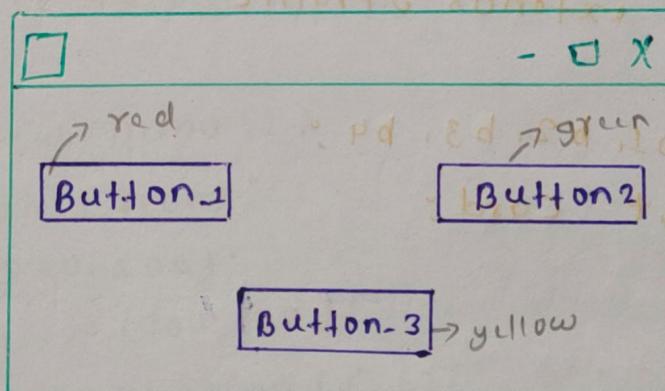
```

if (e.getSource() == b3)
{
    cn.setBackground(Color.YELLOW);
}

class Demo
{
    public static void main (String args)
    {
        FDemo f = new FDemo();
        f.setVisible(true);
        f.setSize(500,500);
        f.setLocation(100,100);
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

```

3 3



CARD LAYOUT

- * Card Layout ka use online exam ke liye kiya jata hai
- * jab bhi ham cardLayout ka use karte hai card ka name bhi de sakte hai
- * Card Layout ke pass Total 5 method hai

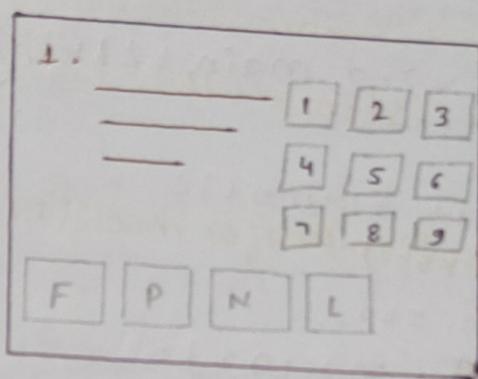
1. **F** - First

2. **P** - Previous

3. **N** - Next

4. **L** - Last

5. **S** - Show



③ import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class FDemo extends JFrame
{
 JButton b1, b2, b3, b4;
 CardLayout card;
 FDemo()
 {
 card = new CardLayout();
 setLayout(card);
 b1 = new JButton("click-1");
 add(b1);
 b2 = new JButton("click-2");
 add(b2);
 b3 = new JButton("click-3");
 add(b3);
 }

b4 = new JButton("click_4");
add(b4);

b1.addActionListener(this);
b2.addActionListener(this);
b3.addActionListener(this);
b4.addActionListener(this);

3
public void actionPerformed(ActionEvent)

{

card.next(cn);

3

3

class Demo

{

public static void main(String args)

{

FDemo f = new FDemo();

f.setVisible(true);

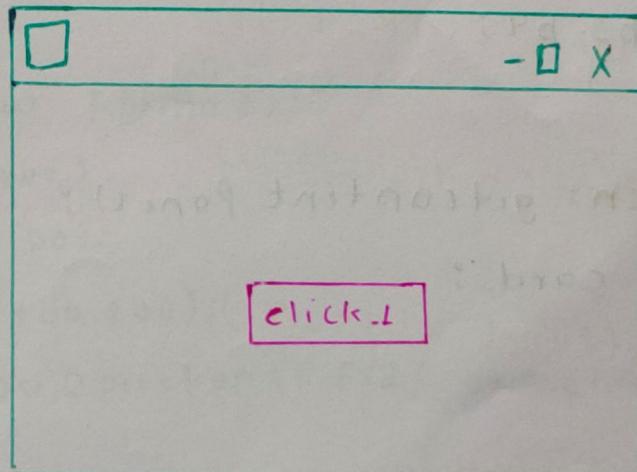
f.setSize(500, 500);

f.setLocation(100, 100);

f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

3

3



1 JPDemo

```
class JPDemo extends JPanel implements ActionListener
{
    JTextField tx1, tx2;
    JButton b1, b2;
    JPDemo2();
    {
        tx1 = new JTextField(10);
        add(tx1);
        b1 = new JButton("click +");
        add(tx2);
        b1.addActionListener(this);
    }
}
```

public void actionPerformed(ActionEvent e)

```
{  
    System.out.println("ram");  
}
```

3
java demo

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class FDemo extends JFrame implements ActionListener
{
}
```

JButton b1, b3, b4;

JPDemo jp;

Container cn = getContentPane();

CardLayout card;

FDemo()

```
{  
}
```

```
card = new CardLayout();  
setLayout(card);
```

```
b1 = new JButton("click-1");  
add(b1);
```

```
jp = new JPanel();  
add(jp);
```

```
b3 = new JButton("click-3");  
add(b3);
```

```
b4 = new JButton("click-4");  
add(b4);
```

```
b1.addActionListener(this);
```

```
b3.addActionListener(this);
```

```
b4.addActionListener(this);
```

3

```
public void actionPerformed(ActionEvent e)
```

```
{
```

```
card.next(en);
```

3

3

```
class p1mo
```

```
{
```

```
public static void main(String args)
```

```
{
```

```
FDemo f = new FDemo();
```

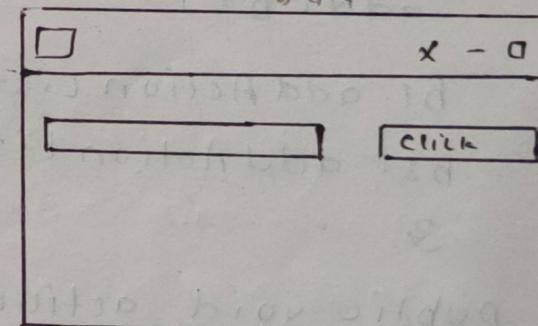
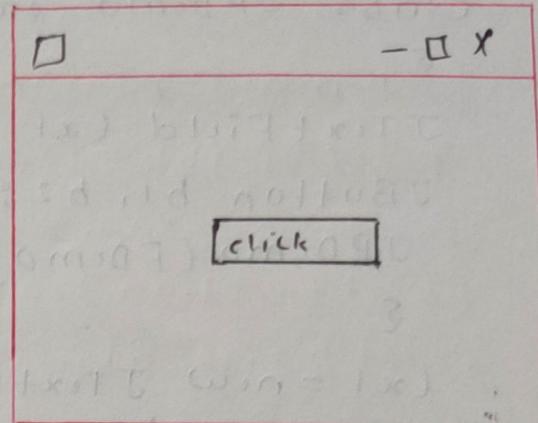
```
f.setVisible(true);
```

```
f.setSize(100, 100);
```

```
f.setLocation(200, 200);
```

```
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

3
y



```

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

class JPDemo extends JPanel implements ActionListener
{
    JTextField tx1, tx2; में परामिटर में
    JButton b1, b2; → FDemo class का object बनाते से
    JPDemo(FDemo f) अर्थात् इस object को action
    { Listener pass करते से
        tx1 = new JTextField(10);
        add(tx1);

        tx2 = new JTextField(10);
        add(tx2);

        b1 = new JButton("click_1");
        add(b1);

        b2 = new JButton("click_2");
        add(b2);

        b1.addActionListener(f);
        b2.addActionListener(this);
    }

    public void actionPerformed(ActionEvent e)
    {
        System.out.println("ram");
    }
}

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

```

class FDemo extends JFrame implements ActionListener

{

 JButton b1, b3, b4;

 JPDemo jp;

 Container cn = getContentPane();

 CardLayout card;

 FDemo()

{

 card = new CardLayout();

 setLayout(card);

 b1 = new JButton("click - 1");

 add(b1);

 jp = new JPDemo(this);
 → इस parameter में this pass करते से
 मे constructor FDemo के
 reference ko hold करेगा

 add(jp);

 b4 = new JButton("click - 4");

 add(b4);

 b5 = new JButton("click - 5");

 add(b5);

 b1.addActionListener(this);

 b4.addActionListener(this);

 b5.addActionListener(this);

 }

 public void actionPerformed(ActionEvent e)

 {

 card.next(cn);

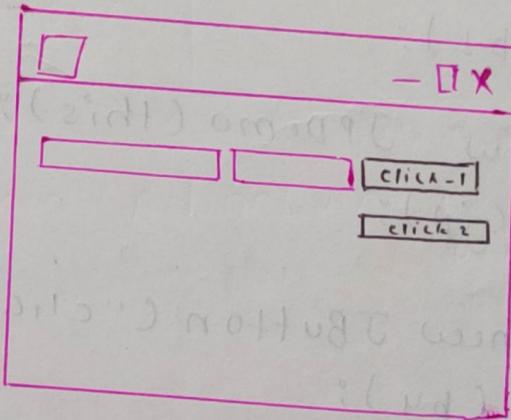
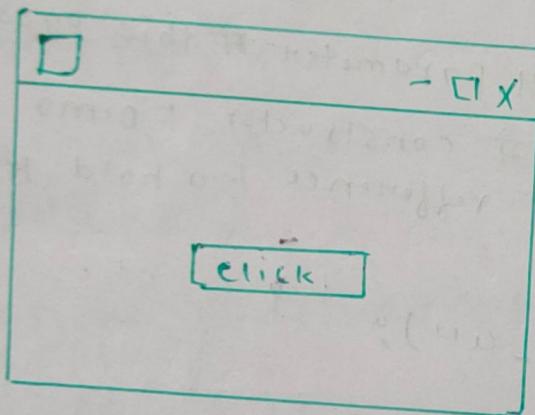
 }

 }

```

class Demo
{
    public static void main (String ar[])
    {
        F Demo f=new FDemo ();
        f.setVisible (true);
        f.setSize (500, 500);
        f.setLocation (100,100);
        f.setDefaultCloseOperation (F.EXIT_ON_CLOSE);
    }
}

```



NOTE:- JP Demo class में JP Demo के parameter में FDemo और उसका object pass करेंगे फिर JP Demo class के Button में Listener में FDemo का object pass कर देंगे और फिर FDemo class के JP Demo (this) Pass करने पर वो FDemo के Listener को hold करे गा

28/07/23
Friday

JPDemo.java

```
import javax.swing.*;  
class JPDemo extends JPanel  
{
```

```
JTextField tx1, tx2;
```

```
JButton b1;
```

```
JPDemo()  
{
```

```
tx1=new JTextField(10);  
add(tx1);
```

```
tx2=new JTextField(10);  
add(tx2);
```

```
b1=new JButton("click....!");
```

```
add(b1);
```

```
}
```

```
}
```

ii) import javax.swing.*;

```
import java.awt.*;
```

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

```
class FDemo extends JFrame implements ActionListener
```

```
{
```

```
JButton b1, b2, b3;
```

```
JPDemo jp;
```

```
Container cn=getContentPane();
```

```
CardLayout card;
```

```
FDemo()  
{
```

```
card=new CardLayout();
```

```
setLayout(card);
```

```
b1=new JButton ("click...1");  
add(b1);
```

```
jP=new JPanel();  
add(jP);
```

```
b3=new JButton ("click...3");  
add(b3);
```

```
b4=new JButton ("click...4");  
add(b4);
```

```
b1.addActionListener(this);  
JP.b1.addActionListener(this);  
b3.addActionListener(this);  
b4.addActionListener(this);
```

3

Public void actionPerformed (ActionEvent e) અર્થ કરો
{

```
Card.next(cn);
```

3
3

```
class Demo
```

{

```
public static void main (String args)
```

{

```
FDemo f=new FDemo();
```

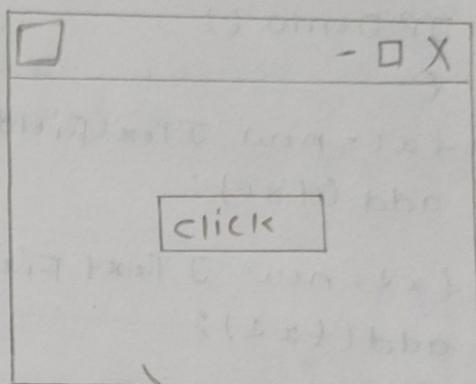
```
f.setVisible (true);
```

```
f.setSize (500, 500);
```

```
f.setLocation (200, 200);
```

```
f.setDefaultCloseOperation (f.EXIT_ON_CLOSE);
```

3
3



```

② import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

class FDemo extends JFrame implements ActionListener
{
    Container cn = getContentPane();
    CardLayout card;
    HomeDemo hd;
    LoginDemo ld;
    Reg Demo rd;

    FDemo()
    {
        card = new CardLayout();
        setLayout(card);

        hd = new HomeDemo();
        add("Home", hd);

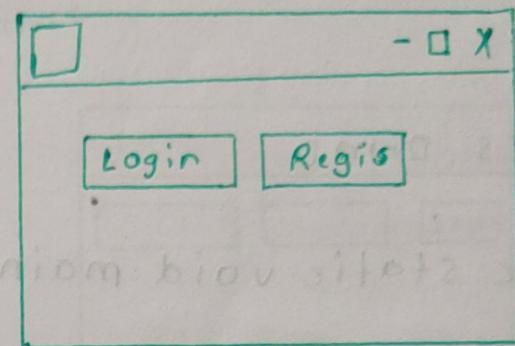
        ld = new LoginDemo();
        add("Login", ld);

        rd = new Reg Demo();
        add("reg", rd);

        hd.b1.addActionListener(this);
        hd.b2.addActionListener(this);
    }

    public void actionPerformed(ActionEvent e)
    {
        if(e.getSource() == hd.b1)
        {
            ...
        }
    }
}

```



```

card.next(cn);
}
if(e.getSource() == b1;b2)
{
card.show(cn, "Reg");
}
}
}
}

class Demo
{
public static void main (String args[])
{
FDemo f=new FDemo();
f.setVisible(true);
f.setSize(500,500);
f.setLocation(200,200);
f.setDefaultCloseOperation(F.EXIT_ON_CLOSE);
}
}

```

HomeDemo

```

import javax.swing.*;
class HomeDemo extends JPanel
{
JButton b1,b2;
HOME Demo()
{
b1=new JButton("Login");
add(b1);
}

```

```
b2=new JButton("Regi");
add(b2);
3
```

Login Demo

```
import javax.swing.*;
class LoginDemo extends JPanel
{
```

```
JTextField tx1,tx2;
```

```
JButton b1;
```

```
>Login Demo()
```

```
{
```

```
tx1=new JTextField(10);
add(tx1);
```

```
tx2=new JTextField(10);
add(tx2);
```

```
b1=new JButton("Login");
add(b1);
```

```
3
```

Reg Demo

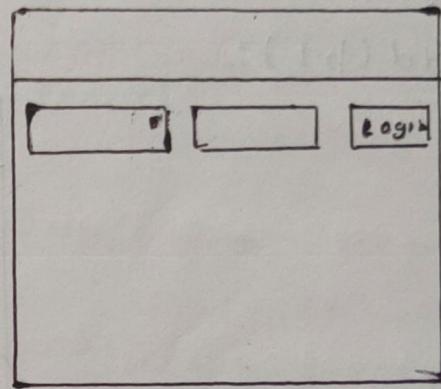
```
import javax.swing.*;
class RegDemo extends JPanel
{
```

```
JTextField tx1,tx2,tx3;
```

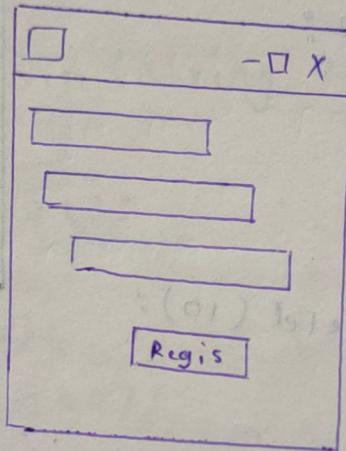
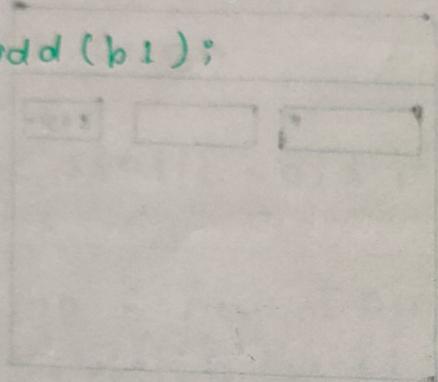
```
JButton b1;
```

```
Reg Demo()
```

```
{
```



```
tx1 = new JTextField(10);  
add(tx1);  
  
tx2 = new JTextField(10);  
add(tx2);  
  
tx3 = new JTextField(10);  
add(tx3);  
  
b1 = new JButton("Registration");  
add(b1);
```



29/07/23
Saturday
31/07/23
Monday

MAIN PROGRAM

```
① import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class FDemo extends JFrame implements ActionListener
{
    CardLayout card;
    Container cn = getContentPane();
    HomeDemo home;
    LoginDemo lg;
    AboutDemo about;
    RegisDemo regis;
    MenuDemo menu;
    FDemo()
    {
        card = new CardLayout();
        setLayout(card);
        home = new HomeDemo();
        add("hm", home);
        lg = new LoginDemo();
        add("login", lg);
        regis = new RegisDemo();
        add("rg", regis);
        menu = new MenuDemo();
        add("menu", menu);
    }
}
```

```
about = new AboutDemo();
```

```
add ("about", about);
```

```
home.b1. addActionListener (this);
```

```
home.b2. addActionListener (this);
```

```
rgis.b2. addActionListener (this);
```

```
lg.b1. addActionListener (this);
```

```
lg.b2. addActionListener (this);
```

3

```
public void actionPerformed (ActionEvent e)
```

```
{
```

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

```
{
```

```
card.show (cn, "login");
```

3

```
if (e.getSource () == home.b2)
```

```
{
```

```
card.show (cn, "rg");
```

3

```
if (e.getSource () == lg.b2 || e.getSource () == rgis.b2)
```

```
{
```

```
card.show (cn, "hm");
```

3

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

```
{
```

```
String s1 = lg.tx1.getText ();
```

```
String s2 = lg.tx2.getText ();
```

```
    }
    card.show(cn, "menu");
}
else
{
    JOptionPane.showMessageDialog(lg, bl, "Invalid Username  
and Password", JOptionPane.ERROR_MESSAGE);
}
```

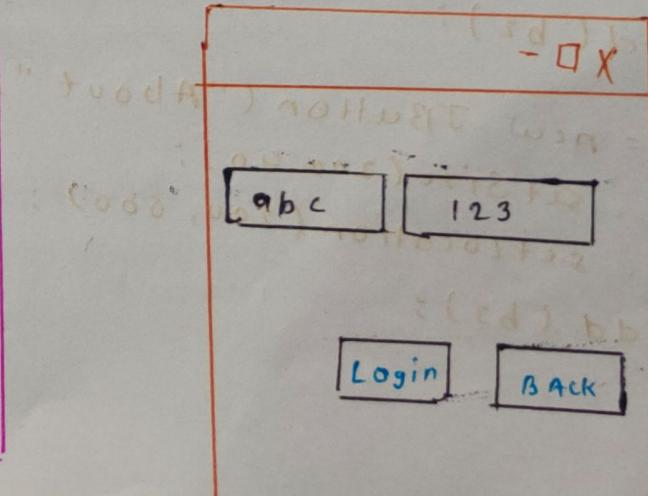
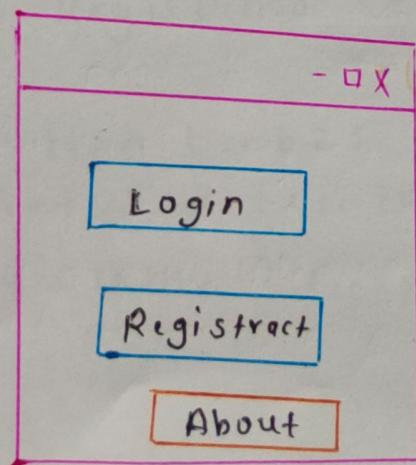
WARNING -
QUESTION -

INFORMATION -

PLAIN

```
class Demo
```

```
{  
    public static void main (String args)  
    {  
        FDemo f=new FDemo();  
        f.setVisible (true);  
        f.setBounds (200,30,800,700);  
        f.setDefaultCloseOperation (f.EXIT_ON_CLOSE);  
    }  
}
```



```
if (s1.equals("abc") & s2.equals("(123"))  
    {  
        card.show(cn, "menu");  
    }  
else  
    {  
        JOptionPane.showMessageDialog(lg, bl, "Invalid user name  
and password", JOptionPane.ERROR_MESSAGE);  
    }
```

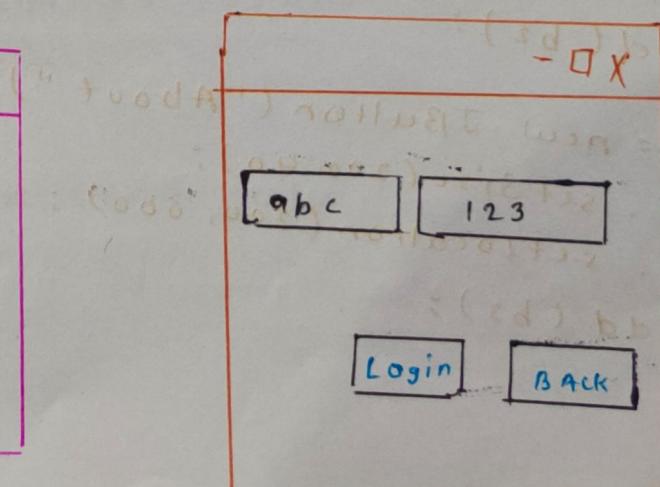
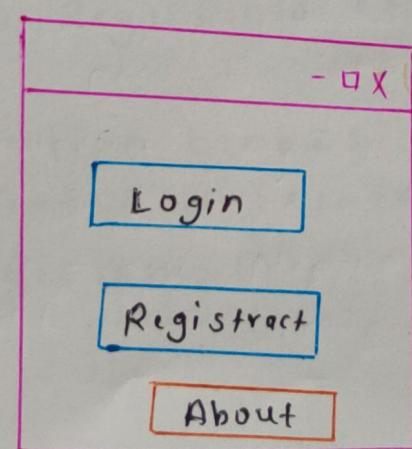
WARNING
QUESTION

INFORMATION

PLAIN

class Demo

```
{  
    public static void main (String args)  
    {  
        FDemo f=new FDemo();  
        f.setVisible (true);  
        f.setBounds (200, 30, 800, 700);  
        f.setDefaultCloseOperation (f.EXIT_ON_CLOSE);  
        f.setDefaultCloseOperation (f.EXIT_ON_CLOSE);  
    }  
}
```



<input type="checkbox"/>	-	<input type="checkbox"/>	X	
INSERT	SEARCH	SHOWALL	UPDATE	DELETE
Enter ROLL NO. _____				
Enter NAME _____				
Enter PHYSICS _____				
Enter CHEMISTRY _____				
Enter MATHS _____				
<input type="button" value="INSERT"/>				

HOME DEMO

```

⑩ import javax.swing.*;
class HomeDemo extends JPanel
{
    JButton b1, b2, b3;
    Home Demo()
    {
        setLayout(null);
        b1 = new JButton("Login");
        b1.setBounds(200, 40, 200, 200);
        add(b1);
        b2 = new JButton("Registration");
        b2.setSize(200, 40);
        b2.setLocation(200, 300);
        add(b2);
        b3 = new JButton("About");
        b3.setSize(200, 40);
        b3.setLocation(200, 600);
        add(b3);
    }
}

```

LOGIN DEMO

```
import javax.swing.*;  
class LoginDemo extends JPanel
```

{

```
    JButton b1, b2;  
    JTextField tx1, tx2;
```

```
    LoginDemo()
```

{

```
    tx1 = new JTextField(10);
```

```
    add(tx1);
```

```
    tx2 = new JTextField(20);  
    add(tx2);
```

```
    b1 = new JButton("Login");  
    add(b1);
```

```
    b2 = new JButton("Back");  
    add(b2);
```

3

3

REGIS DEMO

```
import javax.swing.*;  
class RegisDemo extends JPanel
```

{

```
    JButton b1, b2;  
    JTextField tx1, tx2, tx3;
```

```
    RegisDemo()
```

{

```
tx1 = new JTextField(10);  
add(tx1);  
  
tx2 = new JTextField(10);  
add(tx2);  
  
tx3 = new JTextField(10);  
add(tx3);
```

```
b1 = new JButton("Registration");  
add(b1);  
  
b2 = new JButton("BACK");  
add(b2);
```

3

ABOUT DEMO

```
import javax.swing.*;  
class AboutDemo extends JPanel  
{  
    JButton b1;  
    AboutDemo()  
    {  
        3
```

3

MENU DEMO

```
import java.awt.*;  
import javax.swing.*;  
class MenuDemo extends JPanel  
{  
    3
```

```
Insert Demo ins;
Search Demo syc;
Show All sa;
Update Demo ud;
Delete Demo dd;
JTabbedPane tabl;
JButton b1;
Menu Demo()
{
    setLayout(new BorderLayout());
    tabl = new JTabbedPane();
    ins = new InsertDemo();
    syc = new SearchDemo();
    sa = new SearchAll();
    ud = new UpdateDemo();
    dd = new DeleteDemo();

    tabl.addTab("INSERT", ins);
    tabl.addTab("SEARCH", syc);
    tabl.addTab("SEARCH", sa);
    tabl.addTab("UPDATE", ud);
    tabl.addTab("DELETE", dd);
    add(tabl);
}
3
3
→ class InsertDemo extends JPanel
{
    JLabel u1, u2, u3, u4, u5;
```

```
JTextField f1, f2, f3, f4, f5;
JButton b1;

Insert Demo()
{
    setLayout(null);

    u1 = new JLabel ("Enter ROLL NO.");
    u1.setBounds (100,100, 100,30);
    add(u1);

    f1 = new JTextField(10);
    f1.setBounds (210,105, 200,25);
    add(f1);

    u2 = new JLabel ("Enter NAME");
    u2.setBounds (100,150, 100,30);
    add(u2);

    f2 = new JTextField(10);
    f2.setBounds (210,155, 200,25);
    add(f2);

    u3 = new JLabel ("Enter PHYSICS");
    u3.setBounds (100, 200, 100,30);
    add(u3);

    f3 = new JTextField(10);
    f3.setBounds (210,205, 200,25);
    add(f3);

    u4 = new JLabel ("Enter CHEMISTRY");
    u4.setBounds (100, 250, 100,30);
    add(u4);
}
```

```
t4 = new JTextField(10);
t4.setBounds(200, 200, 100, 30);
add(t4);

l5 = new JLabel("Enter MATH");
l5.setBounds(100, 300, 100, 30);
add(l5);

t5 = new JTextField(10);
t5.setBounds(210, 305, 200, 25);
add(t5);

b1 = new JButton("INSERT");
b1.setSize(100, 200);
b1.setLocation(150, 350);
add(b1);

→ class SearchDemo extends JPanel
{
    JTextField t1;
    JButton b1;
    JLabel l1;
    Search Demo()
    {
        setLayout(null);
        l1 = new JLabel("Enter Name");
        l1.setBounds(100, 100, 100, 30);
        add(l1);
```

```
t1 = new JTextField(10);  
t1.setBounds(200, 110, 200, 25);  
add(t1);  
  
b1 = new JButton("search");  
b1.setBounds(200, 150, 100, 30);  
add(b1);
```

3

3

```
class ShowAll extends JPanel
```

{

```
JTextField t1;
```

```
JButton b1;
```

```
JLabel l1;
```

```
ShowAll().
```

{

```
setLayout(null);
```

```
l1 = new JLabel("Enter Name");
```

```
l1.setBounds(100, 100, 100, 30);
```

```
add(l1);
```

```
t1 = new JTextField(10);
```

```
t1.setBounds(200, 110, 200, 25);
```

```
add(t1);
```

```
b1 = new JButton("search All");
```

```
b1.setBounds(200, 150, 100, 30);
```

```
add(b1);
```

3

3

```
class UpdateDemo extends JPanel
{
    JTextField t1;
    JButton b1;
    JLabel l1;

    UpdateDemo()
    {
        setLayout(null);
        l1 = new JLabel("Enter Name");
        l1.setBounds(100, 100, 100, 30);
        add(l1);

        t1 = new JTextField(10);
        t1.setBounds(200, 110, 200, 25);
        add(t1);

        b1 = new JButton("Update");
        b1.setBounds(200, 150, 100, 30);
        add(b1);
    }

    3
    3

    → class DeleteDemo extends JPanel
    {
        JTextField t1;
        JButton b1;
        JLabel l1;

        DeleteDemo()
        {
            setLayout(null);
        }
    }
```

```
l1 = new JLabel ("Enter Name");  
l1.setBounds (100, 100, 100, 30);  
add(l1);
```

```
f1 = new JTextField (10);  
f1.setBounds (200, 110, 200, 25);  
add(f1);
```

```
b1 = new JButton ("DELETE 1");  
b1.setBounds (200, 150, 100, 30);  
add(b1);
```

3
3

OBJECT CLASS

* java ki sabse super class Object class. baki sohi child class hai.

• Object class ke pass total 12 method hai

1. javap java.lang.Object

RETURN TYPE	METHOD NAME
* class	1. getClass ()
* String	2. toString ()
* int	3. hashCode ()
* boolean	4. equals ()
* void	5. finalize ()
* Object	6. clone
	7. wait
	8. wait (long);
	9. wait (long, int);
	10. notify ();
	11. notifyAll ();

• Object class ke pass 12 method hai

jisme 6 method final hai yeh method ko kahi
bhi override nahi kar sakte hai so they are final

1. getClass

2. wait

3. wait (long);

4. wait (long, int);

5. wait, notify

6. notifyAll

* 2 method protected hai

1. finalize

2. clone

1 private method hai

* register(natives)

11/08/23
Tuesday

there are six step to connect to database :-

1. import package.
2. Load and Register the Driver
3. Opening database connection to MySQL Server.
4. getting statement object to execute query.
5. executing query
6. close connection

1

```
import java.sql.*;  
class Demo  
{  
public static void main(String args) throws Exception  
{  
    Class.forName("com.mysql.jdbc.Driver");  
    Connection con = DriverManager.getConnection("jdbc:  
        mysql://localhost:3306/ajava?  
        useSSL=false", "root", "root");  
    String s1 = "Insert into login values ('abc', '123');";  
    Statement st = con.createStatement();  
    st.executeUpdate(s1);  
    System.out.println("DATA INSERT");  
    con.close();  
}
```

3

3

∴ DATA INSERT

```
② import java.sql.*;
import java.util.*;

class Demo
{
    public static void main (String args) throws Exception
    {
        Scanner sc=new Scanner (System.in);
        System.out.println ("Enter any name and Pass");
        String s1=sc.next();
        String s2=sc.next();
        Class.forName ("com.mysql.cj.jdbc.Driver");
        Connection con=DriverManager.getConnection ("jdbc:mysql://localhost:3306/adjava", "root", "root");
        String s3="insert into login values ('"+s1+"','"+s2+"')";
        Statement st=con.createStatement ();
        st.executeUpdate (s3);
        System.out.println ("data insert");
        con.close ();
    }
}

mysql> show databases;
mysql> use adjava;
mysql> select * from adjava;
mysql> Create table amitdb (UNAME varchar(30), UPASS varchar(30));
```

Query :- Query 2 type hti hoti hai

1. select query
2. non-select query

i) Select Query :- yadi data fetch (search) करने लगती है तो
Select query ka use karege अर्थात् bhi
Select query kibat kijati है st. executeQuery() का
use kare ge

ii) Non-Select Query :- yadi hamne data ko insert
update करना है तो non-select
query ka use kare ge jab bhi non-select query kibat
kijati hai to usme st. executeUpdate() का use kare

UNAME	UPASS	UCN
ram	123	8844
sita	222	9732
gita	486	8976
shree	999	8837

yadi hamare pass database me bhut sara data
rakhा hua hai or ham sita name ke data ko
show करना chahiye hai तो -

→ insert into regis values('sita', '123');
→ Select * from regis where UNAME = 'sita';

sita	222	9732

. ऐसा करते से sita name का jتنا bhi data है sara show hoga.

Select * from regis where UNAME = 'sita' AND UPASS = '321';

SHOW ALL CODE :-

③ import java.sql.*;

class Demo

{

public static void main (String args) throws Exception

{

Class.forName ("com.mysql.cj.jdbc.Driver");

Connection con = DriverManager.getConnection ("jdbc:mysql://localhost:3306/java?useSSL=false",
"root", "root");

String s1 = "select * from login";

Statement st = con.createStatement ();

ResultSet rs = st.executeQuery (s1);

while (rs.next ())

{

System.out.println (rs.getString (1) + " " + rs.getString (2));

}

con.close ();

}

}

Login code

```
④ import java.sql.*;
import java.util.*;
class Demo
{
    public static void main(String args) throws Exception
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter name and Pass");
        String s1=sc.next();
        String s2=sc.next();
        Class.forName("com.mysql.cj.jdbc.Driver");
        Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/adjava556?useSSL=false",
                                                    "root", "root");
        String s3="select * from login where uname='"+s1+
                 "' AND upass='"+s2+"'";
        Statement st=con.createStatement();
        ResultSet rs=st.executeQuery(s3);
        if(rs.next())
        {
            System.out.println("WELCOME");
        }
        else
        {
        }
```

```
System.out.println("invalid user name and pass");
```

3

```
con.close();
```

3

3

* class A

{

```
void show ()
```

{

```
System.out.println("classA");
```

}

3

class Demo

{

```
public static void main (String ar [ ])
```

{

```
A a=new A();
```

```
System.out.println(a);
```

3

* delete from table name

```
delete from regisdemo;
```

* Particular row and column ko delete krne ke liye

```
delete from login where uname = "sita";
```

02/08/23

Wednesday-

① import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;

class RegA extends JPanel implements ActionListener
{
 JButton b1, b2;
 JLabel l1, l2, l3, l4, l5;
 JTextField t1, t2, t3, t4, t5;

 Reg A()
 {
 setLayout(null);
 Font f=new Font("Broadway", Font.BOLD, 20);

 l1=new JLabel("USER NAME");
 l1.setBounds(50, 50, 180, 40);
 add(l1);
 l1.setFont(f);

 t1=new JTextField(10);
 t1.setBounds(200, 50, 180, 40);
 add(t1);
 t1.setFont(f);

 l2=new JLabel("PASSWORD");
 l2.setBounds(50, 120, 180, 40);
 add(l2);
 l2.setFont(f);
 }
}

```
t2 = new JTextField(10);
t2.setBounds(200, 120, 180, 40);
add(t2);
```

```
t3 = new JTextField(10); // JLabel
t3.setBounds(200, 190, 180, 40);
add(t3);
```

```
t3 = new JTextField(10);
t3.setBounds(200, 190, 180, 40);
add(t3);
```

```
t4 = new JLabel("GMAIL");
t4.setBounds(50, 260, 180, 40);
add(t4);
```

```
t4 = new JTextField(10);
t4.setBounds(200, 260, 180, 40);
add(t4);
```

```
b1 = new JButton("REGISTRATION");
add(b1);
b1.setBounds(200, 330, 150, 40);
```

```
b2 = new JButton("BACK");
add(b2);
b2.setBounds(30, 380, 120, 40);
b2.setFont(f);
```

```
b1.addActionListener(this);
```

```
public void actionPerformed(ActionEvent e)
{
    String s1 = t1.getText();
    String s2 = t2.getText();
    String s3 = t3.getText();

    try
    {
        Class.forName("com.mysql.cj.jdbc.Driver");
        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/Rajputdb?", "useSSL=false", "root", "root");

        Statement st = con.createStatement();
        String s1 = "insert into RegA values ('" + s1 + "', '" + s2 + "','" + s3 + "')";
        st.executeUpdate(s1);
        JOptionPane.showMessageDialog(null, "Data insert");
        t1.setText("");
        t2.setText("");
        t3.setText("");

        con.close();
    }
    catch(Exception e1)
    {
    }
}
```

System.out.println(“”);

3

3

3

A hand-drawn diagram of a registration form window. The window has a title bar with a close button (X). It contains three input fields labeled "USER NAME", "PASSWORD", and "CONTRACT", each with a red border. Below the fields are two buttons: "BACK" on the left and "Register" on the right.

mysql> use Rajputdb;

mysql> select * from RegA;

UNAME	UPASS	UCNO
Amit	1234	9617
amit	Rajput123	997788
Ramji	0000	91114

03/08/23

thrusday

Login Demo

①

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

class FDemo extends JFrame implements ActionListener
{
    Container cn = getContentPane();
    CardLayout card;
    HomeA hm;
    LoginA lg;
    RegA rg;
    MenuA md;

    FDemo()
    {
        card = new CardLayout();
        setLayout(card);

        hm = new HomeA();
        add("hm", hm);

        lg = new LoginA();
        add("lg", lg);

        rg = new RegA();
        add("rg", rg);

        md = new MenuA();
        add("md", md);
    }
}
```

```
hm.b1.addActionListener(this);
hm.b2.addActionListener(this);
lg.b1.addActionListener(this);
lg.b2.addActionListener(this);
rg.b2.addActionListener(this);
```

3

```
public void actionPerformed(ActionEvent e)
```

{

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

{

```
card.show(cn,"lg");
```

3

```
if(e.getSource() == hm.b2)
```

{

```
card.show(cn,"rg");
```

3

```
if(e.getSource() == lg.b2 || e.getSource() == rg.b2)
```

{

```
card.show(cn,"hm");
```

3

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

{

```
String s1 = lg.t1.getText();
```

```
String s2 = lg.t2.getText();
```

try

{

```
Class.forName("com.mysql.cj.jdbc.Driver");
```

```
Connection con = DriverManager.getConnection("jdbc:mysql://111 Rajputdb", "root", "root");
```

```
String s11="select *from RegA where UNAME ='" + s1 +
    AND UPASS ='" + s2 + "'";
```

```
Statement st=con.createStatement();
```

```
ResultSet rs=st.executeQuery(s11);
```

```
if(rs.next())
```

```
{  
    card.show(cn,"md");
```

```
}
```

```
else
```

```
{
```

```
JOptionPane.showMessageDialog(null,"invalid user name and  
pass");
```

```
lg.t1.setText(");
```

```
lg.t2.setText(");
```

```
}
```

```
con.close();
```

```
}
```

```
catch(Exception e1)
```

```
{
```

```
System.out.println(e1);
```

```
class Demo
```

```
{
```

```
public static void main (String args)
```

```
FDemo f=new FDemo(); f.setVisible(true); f.setTitle("User Name")
```

```
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

f.setBounds(100, 100, 500, 500);

f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

3

3

A simple window frame with a close button at the top right. Inside, there are two rectangular buttons labeled "LOGIN" and "REGIST".

A window frame with a close button at the top right. It contains fields for "UNAME" and "PASSWORD", and buttons for "BACK" and "Login".

An input form with a title bar showing icons for INSERT, SEARCH, update, showall, and delete. It has five text input fields for "Enter ROLL", "ENTER NAME", "Enter PHY", "Enter CHEM", and "Enter MATH", followed by an "INSERT" button.

URoll	UName	NOPHY	NOCHEM	NONATH
1234	sita	55	55	55

mysql > use Rajputdb;

mysql > select * from marksall;

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

```
class Menu extends JPanel
```

```
{
```

```
JTabbedPane tab1;
```

```
BorderLayout br;
```

```
insert ins;
```

```
Menu()
```

```
{
```

```
br=new BorderLayout();
```

```
setLayout(br);
```

```
tab1=new JTabbedPane();
```

```
ins=new insert();
```

```
tab1.addTab("insert", ins);
```

```
add(tab1);
```

```
3
```

```
class insert extends JPanel implements ActionListener
```

```
{
```

```
JButton b1;
```

```
JLabel l1, l2, l3, l4, l5;
```

```
JTextField t1, t2, t3, t4, t5;
```

```
insert()
```

```
{
```

```
setLayout(null);
```

```
l1 = new JLabel ("Enter Roll No.");  
l1.setBounds (50, 50, 180, 40);  
add(l1);
```

```
t1 = new JTextField (10);
```

```
t1.setBounds (250, 50, 180, 40);  
add(t1);
```

```
l2 = new JLabel ("Enter Name");  
l2.setBounds (50, 110, 180, 40);  
add(l2);
```

```
t2 = new JTextField (10);
```

```
t2.setBounds (250, 110, 180, 40);  
add(t2);
```

```
l3 = new JLabel ("Enter PHY ");
```

```
l3.setBounds (50, 170, 180, 40);  
add(l3);
```

```
t3 = new JTextField (10);
```

```
t3.setBounds (250, 170, 180, 40);  
add(t3);
```

```
l4 = new JLabel ("Enter CHM ");
```

```
l4.setBounds (50, 230, 180, 40);  
add(l4);
```

```
t4 = new JTextField (10);
```

```
t4.setBounds (50, 230, 180, 40);  
add(t4);
```

```
t5 = new JTextField (10);
```

```
t5.setBounds (250, 230, 180, 40);  
add(t5);
```

```
ls = new JLabel("Enter MATH4");
ls.setBounds(50, 290, 180, 40);
add(ls);

b1 = new JButton("Insert");
add(b1);
b1.setBounds(180, 360, 120, 40);

b1.addActionListener(this);
```

3

```
public void actionPerformed(ActionEvent e) {
    if(e.getSource() == b1) {
        String s1 = t1.getText();
        String s2 = t2.getText();
        String s3 = t3.getText();
        String s4 = t4.getText();
        String s5 = t5.getText();

        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con = DriverManager.getConnection("jdbc:mysql://111.rajputdb", "root", "root");

            String s11 = "insert into marksall values ('" + s1 + "'";
            s11 += "' + s2 + "','" + s3 + "','" + s4 + "','" + s5 + "')";
```

```
Statement st = con.createStatement();
st.executeUpdate(sql);
con.close();

JOptionPane.showMessageDialog(null, "insert data");

f1.setText("");
f2.setText("");
f3.setText("");
f4.setText("");
f5.setText("");

catch (Exception e1)
{
    System.out.println(e1);
}
```

03/08/23

thrusday

SEARCH DEMO

```
class Search extends JPanel implements ActionListener
{
    JButton b1;
    JLabel l1, l2, l3, l4, l5, l6, l7, l8, l9, l10, l11;
    JTextField t1;

    Search()
    {
        setLayout(null);
        Font f=new Font ("",Font.BOLD,15);
        l1=new JLabel ("Enter ROLL NO");
        l1.setBounds (20,50,180,40);
        add(l1);

        t1=new JTextField (10);
        t1.setBounds (200,50,180,40);
        add(t1);

        l2=new JLabel ("ROLL NO");
        l2.setBounds (20,150,180,40);
        add(l2);

        l3=new JLabel ("NAME");
        l3.setBounds (140,150,180,40);
        add(l3);

        l4=new JLabel ("PHYSICS");
        l4.setBounds (240,150,180,40);
        add(l4);
    }
}
```

```
l5 = new JLabel ("CHEMISTRY");  
l5 . setBounds (340, 150, 180, 40);  
add (l5);
```

```
l6 = new JLabel ("MATH");  
l6 . setBounds (480, 150, 180, 40);  
add (l6);
```

```
l7 = new JLabel ("");  
l7 . setBounds (20, 200, 180, 40);  
add (l7);
```

```
l8 = new JLabel ("");  
l8 . setBounds (140, 200, 180, 40);  
add (l8);
```

```
l9 = new JLabel ("");  
l9 . setBounds (240, 200, 180, 40);  
add (l9);
```

```
l10 = new JLabel ("");  
l10 . setBounds (340, 200, 180, 40);  
add (l10);
```

```
l11 = new JLabel ("");  
l11 . setBounds (480, 200, 180, 40);  
add (l11);
```

```
b1 = new JButton ("SEARCH");  
b1 . setBounds (410, 50, 120, 40);  
add (b1);
```

```
b1.addActionListener(this);  
3  
public void actionPerformed(ActionEvent e)  
{  
    String s1 = t1.getText();  
    try  
    {  
        Class.forName("com.mysql.cj.jdbc.Driver");  
        Connection con = DriverManager.getConnection("jdbc:  
mysql://111rajputdb", "root", "root");  
        Statement st = con.createStatement();  
        String s2 = "select * from marksAll where UR011  
= " + s1 + " ";  
        st.executeQuery(s2);  
        ResultSet rs = st.executeQuery(s2);  
        if (rs.next())  
        {  
            t7.setText(rs.getString(1));  
            t8.setText(rs.getString(2));  
            t9.setText(rs.getString(3));  
            t10.setText(rs.getString(4));  
            t11.setText(rs.getString(5));  
        }  
        else  
        {  
    }
```

```
JOptionPane.showMessageDialog(null, INVALID  
VALID ROLL NO");
```

```
l7. setText("");  
l8. setText("");  
l9. setText("");  
l10. setText("");  
l11. setText("");
```

```
3  
con.close();
```

```
3  
catch (Exception e1)
```

```
5  
System.out.println(e1);
```

```
3  
3  
3
```

```

② class Employee
{
    void show()
    {
        System.out.println("class A");
    }
}

class Demo
{
    public static void main(String args[])
    {
        Employee e=new Employee();
        System.out.println(e);
        System.out.println(e.getClass());
        System.out.println(e.getClass().getName());
    }
}

```

Output - Employee@7852e922

class Employee

Employee

NOTE: jab kabhi bhi ham kisi bhi object ko print karte hai to output className @ hexa decimal no. ke form me hi aaye ga

* jab kabhi bhi ham kisi bhi object ko print karte hai to us object ke corresponding to string method call hoti hai;

③ class Employee

{

void show()

{

System.out.println("class A");

}

public String toString()

{

return "Ram"; Yaha par toString() method Ram
return kary rahi hai.

}

3

3

class Demo

{

public static void main(String args[])

{

Employee e = new Employee();

System.out.println(e);

System.out.println(e.toString());

3

3

O/p- Ram
Ram

④ class Employee // extends Object

{

void show()

{

System.out.println("class A");

3

3

class Demo

{

```
public static void main(String args)
{
    Employee e=new Employee();
    System.out.println(e);
    System.out.println(e.toString());
}
```

Output - Employee@H.D.N

Employee@H.D.N

05/08/23
Saturday

① class Employee

{

int id;

String add;

void get(int id, String add)

{

this.id = id;

this.add = add;

}

void show()

{

System.out.println("id=" + id);

System.out.println("add=" + add);

}

3

class Demo

{

public static void main(String args[])

{

Employee ram = new Employee();

ram.get(101, "indore");

Op - id = 101

indore

Employee sita = new Employee();

sita.get(102, "indore");

id = 102

add = bhopal

ram.show();

sita.show();

3

3

NOTE: Constructor ki help se value assign kar sakte hai jaise hi object create kiya base hi assign ho gaya

② class Employee

{

int id;

String add;

Employee(int id, String add)

{

this.id = id;

this.add = add;

}

void show()

{

System.out.println("id=" + id);

System.out.println("add=" + add);

}

class Demo

{

public static void main(String args[])

{

Employee ram = new Employee(101, "bhopal");

Employee sita = new Employee(102, "indore");

ram.show();

sita.show();

Output - id = 101

 add bhopal

 id = 102

 add indore

}

3

③ class Employee

```

    {
        int id;
        string add;
        Employee(int id, string add)
        {
            this.id=id;
            this.add=add;
        }
        public String toString()
        {
            return "id=" + id + " " + add;
        }
    }

```

class Demo

```

    {
        public static void main(String args)
        {
            Employee ram=new Employee(101,"bhopal");
            Employee sita=new Employee(102,"indore");
            System.out.println("ram");
            System.out.println("sita");
        }
    }

```

* 300 HAN *

NOTE: Object class ki toString method ko override kar sakte hai ya nahi
YES kar sakte hai

- Override karne se kya benefit hai?
 - I Meaningful output ko display karne ke purpose se toString method ka use karte hai

Jab bhi ham kisi bhi object ko print karte hai to is condition me us object ke corresponding toString method call hoti hai

yadi object ke corresponding toString method nahi he to object class ki toString method call hoti hai or output \Rightarrow class name @ H.D.N ke form me return (Hexa decimal no.) karta hai

class name @ Hexa decimal NO object class ki toString method jo output ko return karta he wo output hamare liye meaningless hai

kisi bhi object ko print karte time meaningful output ko display karne ke purpose se toString method ko override kiya jata hai

or jo hexadecimal number hai usc hi hashCode kaha jata hai i.e. class name @ 124ebe78

↑
hashCode

* HASH CODE *

* Har object ke corresponding JVM ek unique number assign karta hai usi number ko hashCode kahte hai or return type int hai

Object class ki toString method ko hexadecimal number ke form me return karta hai

* Object class ki hashCode method ^{hashcode} r k decimal number ke form me return karta hai

```
class Employee
```

```
{
```

```
}
```

```
class Demo
```

```
{
```

```
public static void main (String args)
```

```
{
```

```
Employee e = new Employee();
```

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

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

```
System.out.println(Integer.toHexString(e.hashCode()));
```

```
e
```

```
Output - Employee @ 7852e922
```

```
2018699554
```

```
7852e922
```

Object class की toString () method का internal coding

मिलता है ?

```
* public String toString()
```

```
{
```

```
return getClass().getName() + "@" + Integer.toHexString(toString());
```

```
e
```

07/08/23

Monday -

① class A

{

void show()

{

System.out.println("class A");

}

}

class B extends A

O/P - class A

{

}

class Demo

{

public static void main(String args[])

{

B b=new B();

b.show();

}

② class A

{

void show()

{

System.out.println("class A");

}

}

class B extends A

{

void show()

{

```
System.out.println("class B");
class Demo
{
    public static void main(String args[])
    {
        B b=new B();
        b.show();
    }
}
```

O/p - class B

METHOD OVERRIDE :- Method ki definition ko change karne ke purpose se method ko override karte hai.

* Har ek object ke corresponding hash code ek unique no provide karta hai

③ class Employee

{

}

class Demo

{

```
public static void main(String args[])
{
    Employee e1=new Employee();
    Employee e2=new Employee();
}
```

{

Employee e1=new Employee();

Employee e2=new Employee();

```
System.out.println(e1.hashCode());
System.out.println(e2.hashCode());
}
```

int st=0
olp class name @ H.P.
@ H.D.N
* 2018 699 554

* 1311053135

NOTE : hashCode() method ham hexa decimal no. convert karke dega

* hashCode method override is condition per 2 object se ham hashCode ko print kar raha hai to hashCode duplicate o/p - de dega

④ class Employee

{

public int hashCode()

{

return 50;

}

class Demo

{

public static void main (String ar[])

{

Employee e1 = new Employee();

Employee e2 = new Employee();

System.out.println(e1.hashCode());

System.out.println(e2.hashCode());

3

3

Q kya ham hashCode method ko override kar sakte
hai ya nahi

* YES kar sakte hai

⑤

class Employee

{

public int hashCode()

{

return 100;

3

3

```
class Demo
{
    public static void main(String args[])
    {
        Employee e1=new Employee();
        System.out.println(e1);
        System.out.println(e1.hashCode());
    }
}
```

System.out.println(e1);
System.out.println(e1.hashCode());

3

3

* Object class ki toString method hashCode ko classname
@ H.D.N ke form me return karta hai

⑥ class Employee

{

int id;

String add;

Employee(int id, String add)

{

this.id=id;

this.add=add;

3

public int hashCode()

{

return id;

3

3

class Demo

{

public static void main(String args[])

{

```
Employee e1=new Employee(101, "indore");
Employee e2=new Employee(102, "bhopal");
```

```
System.out.println(e1);
System.out.println(e2);
```

3
3

16	101		16	102	
16	6	5	16	6	6
	0	6		0	6

⑦ class Employee

{

int id;

String add;

Employee(int id, String add)

{

this.id=id;

this.add=add;

3

public String toString()

{

return ""+id;

3

3

class Demo

{

public static void main (String args[])

{

```
Employee e1=new Employee(101,"indore");
Employee e2=new Employee(102,"bhopal");
```

```
System.out.println(e1);
System.out.println(e2);
```

```
Employee e1=new Employee(101, "indore");
Employee e2=new Employee(102, "bhopal");

System.out.println(e1);
System.out.println(e2);
```

Output - Employee@ 65
Employee@ 66

3
3

16	101		16	102	
16	6	5	16	6	6
0	6		0	6	

⑦ class Employee

{

int id;

String add;

Employee(int id, String add)

{

this.id=id;

this.add=add;

}

public String toString()

{

return ""+id;

}

3

class Demo

{

public static void main(String args)

{

Employee e1=new Employee(101, "indore");

Employee e2=new Employee(102, "bhopal");

```
System.out.println(e1);
System.out.println(e2);
3
3
```

⑧ class Employee

```
{
```

```
int id;
```

```
String add;
```

```
Employee(int id, String add)
```

```
{
```

```
this.id=id;
```

```
this.add=add;
```

```
}
```

```
public int hashCode()
```

```
{
```

```
return id;
```

```
}
```

```
public String toString()
```

```
{
```

```
return getClass().getName() + "@";
```

```
}
```

```
class Demo
```

```
{
```

```
public static void main(String args)
```

```
{
```

```
Employee e1=new Employee(101,"indore");
```

```
Employee e2=new Employee(102,"bhgpal");
```

```
System.out.println(e1);
System.out.println(e2);
```

3
3

⑨ class Employee

{

int id;

String add;

Employee(int id, String add)

{

this.id=id;

this.add=add;

}

public int hashCode()

{

return id;

}

public String toString()

{

return getClass().getName() + "@" + hashCode();

3
3

class Demo

{

public static void main(String args)

{

Employee e1=new Employee(101,"bhopal");

Employee e2=new Employee(102,"indore");

System.out.println(e1);
System.out.println(e2);

3

3

⑩ class Employee

{

int id;

String add;

Employee(int id, String add)

{

this.id=id;

this.add=add;

3

public int hashCode()

{

return id;

3

class Demo

{

public static void main(String ar[])

{

Employee e1=new Employee(101,"bhopal");

Employee e2=new Employee("102","indore");

System.out.println(e1);

System.out.println(e2);

Output- Employee@65

3

3

Employee@66

```
System.out.println(e1);
```

```
System.out.println(e2);
```

3
3

⑨ class Employee

{

int id;

String add;

Employee(int id, String add)

{

this.id = id;

this.add = add;

3

public int hashCode()

{

return id;

3

public String toString()

{

return getClass().getName() + "@" + hashCode();

3
3

class Demo

{

public static void main(String args)

{

Employee e1 = new Employee(101, "bhopal");

Employee e2 = new Employee(102, "indore");

obj- Employee @101

Employee @102

11 class Employee

{

 int id;

 String add;

Employee(int id, String add)

{

 this.id=id;

 this.add=add;

}

public int hashCode()

{

 return id;

}

public String toString()

{

 return getClass().getName()+"@"+Integer.toHexString(hashCode());

}

3

class Demo

{

 public static void main(String args)

 {

 Employee e1=new Employee(101,"bhopal");

 Employee e2=new Employee(102,"indore");

 System.out.println(e1);

 System.out.println(e2);

 }

08/08/23
Tuesday

UPDATE YOUR DATA ON DATABASE

① class update extends JPanel implements ActionListener

 {

 JLabel l1, l2, l3, l4, l5;
 JTextField t1, t2, t3, t4, t5, t6;
 JButton b1, b2;

 update();

 {

 l1 = new JLabel("Enter Roll No");
 l1.setBounds(50, 50, 120, 40);
 add(l1);

 t1 = new JTextField(10);
 t1.setBounds(200, 50, 130, 30);
 add(t1);

 l2 = new JLabel("Enter Name");
 l2.setBounds(50, 100, 120, 40);
 add(l2);

 t2 = new JTextField(10);
 t2.setBounds(200, 100, 130, 30);
 add(t2);

 l3 = new JLabel("Enter Phy");
 l3.setBounds(50, 150, 120, 40);
 add(l3);

 t3 = new JTextField(10);
 t3.setBounds(200, 150, 130, 30);
 add(t3);

```
l4=new JLabel("Enter chem");
l4.setBounds(50, 250, 120, 40);
add(l4);

t4=new JTextField(10);
t4.setBounds(200, 200, 150, 30);
add(t4);

l5=new JLabel("Enter Math");
l5.setBounds(50, 250, 120, 40);
add(l5);

t5=new JTextField(10);
t5.setBounds(200, 250, 150, 30);
add(t5);

b1=new JButton("search");
b1.setBounds(300, 300, 120, 30);
add(b1);

b2=new JButton("UPDATE");
b2.setBounds(300, 350, 120, 30);
add(b2);

b1.addActionListener(this);
b2.addActionListener(this);

public void actionPerformed(ActionEvent e)
{
    if(e.getSource()==b1)
    {
        String s1=t4.getText();
        try
        {
```

```
Class.forName("com.mysql.cj.jdbc.Driver");
Connection con=DriverManager.getConnection ("jdbc:mysql://127.0.0.1:3306/test", "root", "root");
```

```
Statement st=con.createStatement();
String s="select * from mark where UROLL='"+st+"";
```

```
ResultSet rs=st.executeQuery(s);
```

```
if(rs.next())
```

```
{
```

```
t1.setText (rs.getString(1));
```

```
t2.setText (rs.getString(2));
```

```
t3.setText (rs.getString(3));
```

```
t4.setText (rs.getString(4));
```

```
t5.setText (rs.getString(5));
```

```
else
```

```
{
```

```
JOptionPane.showMessageDialog(null,"invalid roll");
```

```
t1.setText("0","invalid");
```

```
t2.setText("0","invalid");
```

```
t3.setText("0","invalid");
```

```
t4.setText();
```

```
t5.setText();
```

```
3
```

```
catch (Exception e)
```

```
{
```

```
System.out.println("1");
3
if (e.getSource() == b2)
{
    String s1 = f1.getText();
    String s2 = f2.getText();
    String s3 = f3.getText();
    String s4 = f4.getText();
    String s5 = f5.getText();
    try
    {
        Class.forName("com.mysql.jdbc.Driver");
        Connection con = DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/adjava", "root", "root");
        Statement st = con.createStatement();
        String s = "update mark set UROLL = '" + s1 + "', UNAME = '" + s2 + "', UPHY = '" + s3 + "', UCHM = '" + s4 + "', UMAH = '" + s5 + "'"
                + " where UROLLNO = '" + s1 + "'";
        st.executeUpdate(s);
        JOptionPane.showMessageDialog(null, "UPDATE");
        f1.setText("");
        f2.setText("");
        f3.setText("");
    }
}
```

```
+4. setText("");  
+5. setText ("");  
}  
catch (Exception e)  
{  
    System.out.println(e);  
}  
}
```

* APACHE TONCAT *

FileName- ServerDemo.java

```
import java.io.*;  
import javax.servlet.*;  
import javax.servlet.http.*;  
  
public class ServerDemo extends HttpServlet  
{  
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws  
        ServletException, IOException  
    {  
        PrintWriter out=response.getWriter();  
        out.println("<html>");  
        out.println("  <body>");  
        out.println("    <h1>RAM</h1>");  
        out.println("  </body>");  
        out.println("</html>");  
    }  
}
```

out.close();

3

3

Program me class name or
file name same rakhna pad
ga

Output :- ERROR

1. import javax.servlet.*;

2. import javax.servlet.http.*;

इस package में 4 class होती हैं।

3. HttpServlet

4. HttpServletRequest

5. HttpServletResponse

6. ServletException

NOTE: इसके program me jis class ko public declare
kiya hai to us condition me hamko file
or class ka name * same rakhna hoga

① public class Demo

{

public static void main(String args)

{

System.out.println("class Ram");

Output - class Ram

②

public class Demo100

{

public static void main(String args)

{

System.out.println("class Ram");

javac Demo 59.java

java Demo 100

ERROR: class demo100 is public should be declare in file name demo100.java

NOTE: ham program me ek se jyada public class dikt sakk hai ya nahi hamare file me

Ans - no.

③ public class A
 { 3 }

public class Demo59 :
 {
 public static void main (String ar^g)
 {
 System.out.println ("class A");
 }
 }

OIP- ERROR: class A is public
should be declared in
file name: A.java public
n
class A

④ class A
 { 3 }

public class demo59
 {
 public static void main (String ar^g)
 {
 System.out.println ("class A");
 }
 }

OIP - class A

INSTALLATION APACHE Tomcat

① Download version: 8.5.64

② then Install Program Files



Apache Software Foundation



Tomcat 8.5



lib → servlet-api →
servlet-api ← उन दोनों package के code
इस jar file में रखा
है।

classpath :: set classpath=c:\programFiles\Apache Software
Foundation\Tomcat 8.5\lib\servlet-api.
jar ;.; %classpath%

* path इसलिए set कर रहे हैं तो program में error
generate हो रही है।

09/08/23
Wednesday

Method Overriding

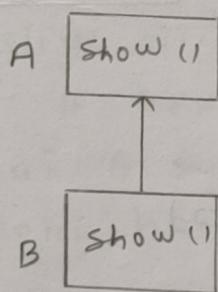
- * Run time polymorphism ka example hai.
- * Method overriding me both class me "instance method" hona chahiye.
- * Method overriding yeh object ke corresponding work karta hai.

Method Hiding

- * Compile time polymorphism ka example hai.
- * Both class static hona chahiye.
- * Reference variable ke corresponding work karta hai.

Super class ka reference variable sub class ki object ko hold karke rakh sakte hai.

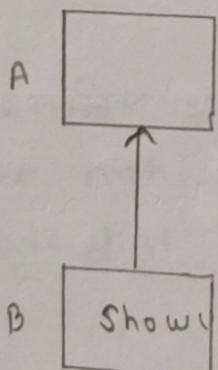
Case 1:



A a=new B();
a.show();

class B

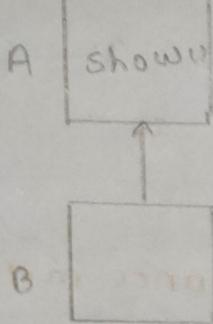
2.



A a=new B();
a.show();

O/P - error: incompatible type; B cannot
be converted to A

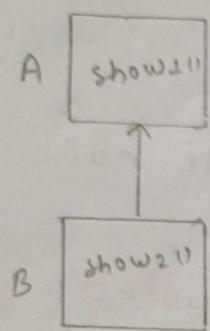
3.



A `a=new B();`
a.show();

OP: class A (inheritance apply)

4.

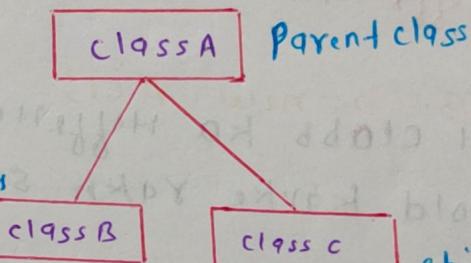


A `a=new B();`
a.show1();
a.show2();

OP: ERROR: Incompatible type B cannot be converted to A

class A
{}
{}
{}
{}

CASE 1:



class B extends A
{}
{}
{}
{}

class C extends A
{}
{}
{}
{}

class Demo
{}
{}
{}
{}

public static void main(String args)

{

A `a=new B();`

A `a2=new C();`

OP: Blank screen

3

A `a=new B();` — object



Reference Variable

* Compile time par checking hamseba reference variable ke corresponding hoti hai.

* Runtime par working hamseba object ke corresponding hoti hai.

* 1. A a1=new B(); ये compile run ho jayega
A a2=new C(); program execute hone me koi gft problem nahi aata
जैसे B वाले C को एक class A की subclass है

OR super class ka reference variable subclass ke object ko hold kar sakte hai.

2. B b=new A(); error: incompatible type A cannot be converted to B

error: istiye आयेरी स्पष्टीय class B का reference variable नहीं है वह hamne object class A ka banaya hai A super class hai or to subclass से convert nahi ho sakti hai

③ B b1=b2=new C(); error: C cannot be convert B

error: इसलिए आयेरी स्पष्टीय class B or class C me koi relation nahi hai to bo ek dusre me convert nahi ho sakte hai

④ A a=new A(); error: incompatible type A
B b=a; cannot be converted to B

error इसलिए aayegi क्योंकि reference variable bhi class A का है और object bhi class A का है या हम class A के reference ko B में store nahi kara sakte क्योंकि A superclass है या B subclass

⑤ A a=new B(); error: incompatible types A
B b=a; cannot be converted to B

error इसलिए आएगी क्योंकि class A का reference variable B में store nahi ho saktा है।

⑥ A a=new C(); error: incompatible type A cannot
B b=a; be converted to B

error इसलिए आएगी क्योंकि reference variable A का है या हम B me store kar rahe हैं

⑦ A a=new A(); Exception :- ClassCastException
B b=(B)a; : A can't be cast to B

compile से ही जाएगी पर runtime पर exception aayegi reference variable class A का है या B me store kar hai या kabhi bhi subclass से superclass में type casting nahi hoti है।

⑧ A a=new B();
B b=(B) a

→ यहाँ से भी object bhi B class का हो OR ham store
bhi class B नहीं हो सकते हैं।

⑨ A a=new C(); Exception : classcast Exception : c
B b=(B) a; cannot be cast to B

exception aayegi क्योंकि C के object ko B hold nahi
kar sakte hai

⑩ C c=new C(); error: incompatible type c cannot
B b=(B) c; be converted to B

इसलिए error आयेगी क्योंकि type casting ऐसा parent
child relation नहीं हो सकती है otherwise nahi hogio

10/08/23

thursday

APACHE TOMCAT

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class SERVLET DEMO extends HttpServlet
{
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException
```

```
PrintWriter out = response.getWriter();
out.println("<html>");
out.println("<body>");
out.println("<h1>RAM</h1>");
out.println("</body>");
out.println("</html>");
out.close();
```

3
3

error: package javax.servlet does not exist
import javax.servlet.*;

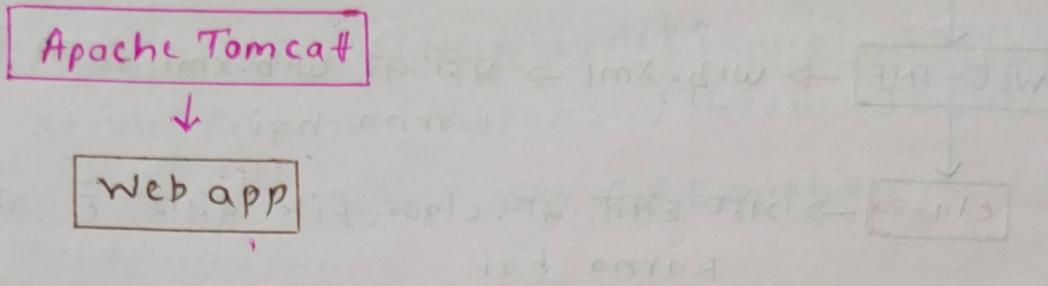
error: package javax.servlet.http does not exist
import javax.servlet.http.*;

error: cannot find symbol
public class SERVLET DEMO extends HttpServlet

error: cannot find symbol
public void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException ServletException

Apache Tomcat → के अंदर एक folder मिलेगा **lib** नाम का उसके अंदर **servlet-api.java**

Set classpath = H:\Fl\apache-tomcat-8.5.64\lib\servlet-api.jar ; . ; %classpath%



* इसी File **web.xml** इस नाम से save karna hai

इसको कहा जाता है Deployment descriptor
इसका use security purpose से हिस्ता बताता है।

* **web.xml**

<web-app>

< servlet >

< servlet-name > amit < /servlet-name >

< servlet-class > ServerDemo1 < /servlet-class >

</servlet >

< servlet-mapping >

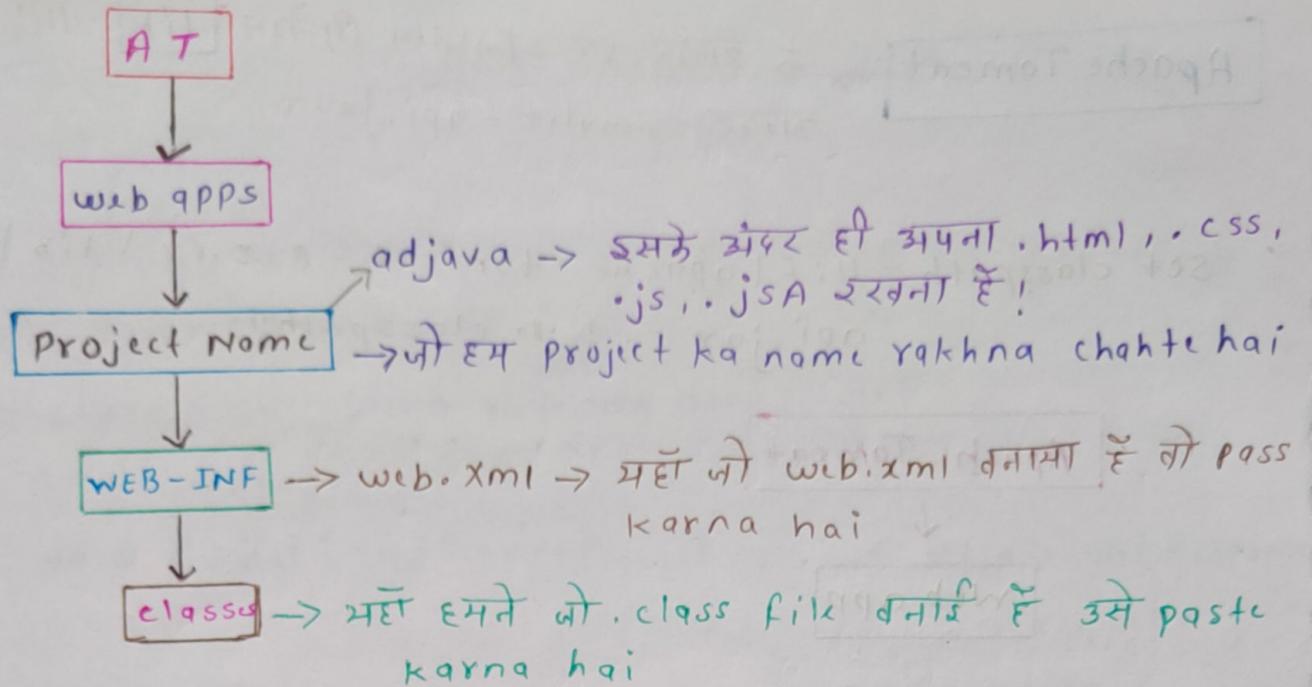
< servlet-name > amit < /servlet-name >

< url-pattern > /abc < /url-pattern >

< /servlet-mapping > ऐसे जो pass करेंगे उसी से run hoga

</web-app>

∴ Program ka original name है Server
Demo1 लेकिन open /abc नाम से hogा
for security purpose



http://localhost:8080/adjava/abc

run hone ka point

Apache Tomcat → bin → startup → shutdown

Query String Concept :- A query string commonly includes fields added to a base URL by a web browser or other client application.

For eg: as a part of HTML choosing the appearance of a page or jumping to positions in multimedia content.

A set of characters tacked onto the end of url

यह इस चाहते हैं url से data मार्क की:-

like - localhost : 8080 / adjava / wel ? un = ramji

o/p - ramji

* इसी concept को query string concept कहते हैं!

Data fetch karne ke liye home request object ka use karna padega.

SERVLET DEMO 2.java

② import java.io.*;
import javax.servlet.*;
import java.servlet.http.*;
public class SERVLET DEMO extends HttpServlet
{
 public void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException
 ServletException
 {
 String str = request.getParameter("un");
 PrintWriter out = response.getWriter();
 out.println("<html>");
 out.println("<body>");
 out.println("<h1>welcome " + str + "</h1>");
 out.println("</body>");
 out.println("</html>");
 out.close();
 }
}

web.xml → यह web.xml का सिर्फ़ code करना पड़ेगा

<web-app>
<servlet>
<servlet-name> abhi2 </servlet-name>
<servlet-class> SERVLET DEMO 2 </servlet-class>
</servlet>

```
<servlet-mapping>
<servlet-name>abhi2 </servlet-name>
<url-pattern>/wel </url-pattern>
</servlet-mapping>
</web-app>
```

run = http://localhost:8080/adjava/wel?un=ramji
↓
टेट जो भी output
प्राइम बी message लिए हो

Deployment descriptor-

three term of create web.xml file

1. Servlet name
2. Servlet class
3. URL pattern

A deployment descriptor for a web application is a XML file that controls how a web application server is deployed (made available to file). For Apache Tomcat, the deployment descriptor file is named web.

- * Object ki default value null hoti hai
- * String ki default value null hoti hai

getParameter :: Return the value of a request parameter as a string or null if the parameter does not exist.
Request parameter all extra information sent with the request. For HTTP servlet, parameters are contained in the query string as posted form data (Run Query se jobhi? un=me jo bhi pass karne ke baare me return ho jayega otherwise null return hogा)

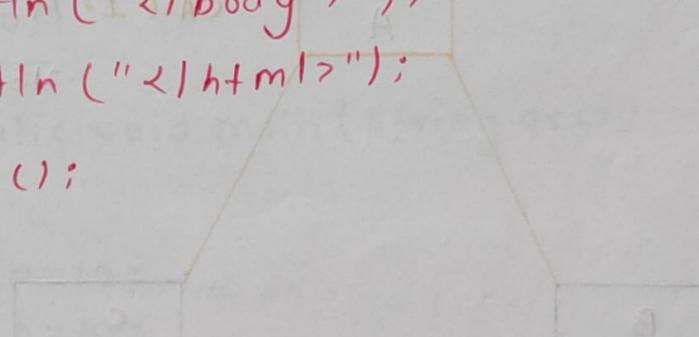
Welcome ramji // Welcome null

11/08/23

Friday

① import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class SERVERDemo4 extends HttpServlet
{
public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException
{
PrintWriter out = response.getWriter();
String s1 = request.getParameter("un");
String s2 = request.getParameter("up");
out.println("<html>");
out.println("<body>");
out.println("<h1> UNAME "+s1+" </h1>");
out.println("<h1> UPASS "+s2+" </h1>");
out.println("</body>");
out.println("</html>");
out.close();
}

3



<web-app>

<servlet>

<servlet-name> amit2 </servlet-name>

<servlet-class> SERVERDemo4 </servlet-class>

</servlet>

```

< servlet-mapping >
< servlet-name > amit2 </ servlet-name >
< url-pattern > /lg </ url-pattern >
</ servlet-mapping >

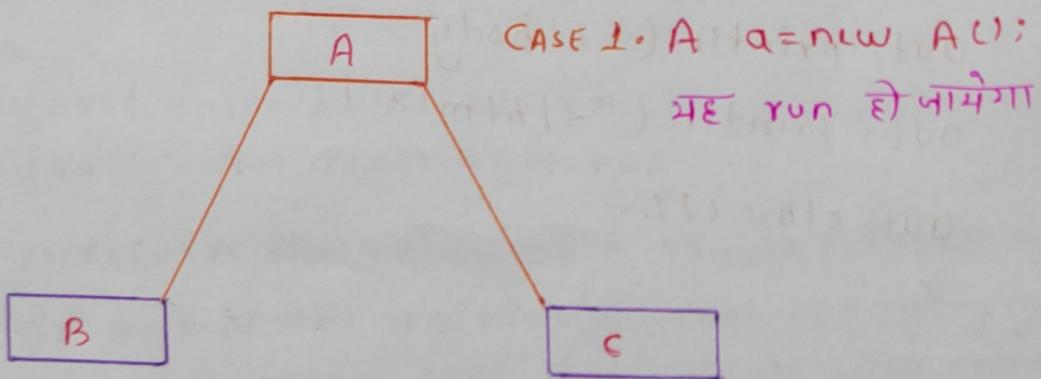
```

`http://localhost:8080/adjava/lg?un=ram&up=123`

Explain class cast Exception

class cast Exception in java is one of the Unchecked Exception that occur when we try to convert one class type object into another class type.

ClassCast Exception is thrown when we try to cast an object of the parent class to the child class object. However it can also be thrown when we try to convert the object of two individual classes that don't have any relationship between them.



Type casting hameba parent child में ही hoti hai

CASE 2. B b = (B) a;

इस condition पर classCastException ayege

CASE : 3

A a=new B();

B b=a;

A a=new B(); *proper work karega*
B b=(B)a;

CASE 4:

A a=new C();

B b=a;

A a=new C(); class cast Exception

B b=(B)a; *अपेक्षित + की तरह c का object*

b के पास नहीं आयेगा।

Run time पर jab different class ka reference variable
different class ke object ko store karta hai or bo
object uska child na ho to is condition par class
cast exception aati hai

② class Demo

{

public static void main(String args)

{

Integer a=10;

Integer b=10;

Integer c=1000;

Integer d=1000;

Integer x=50;

Integer y=50;

Integer p=500;

Integer q=500;