

Notes

Dt: 12 Feb 2018

Tricks

int num; // can't access members of class B.

{ return num++;

to
any
of

(possible)

void getNum(int n)

{ num = n;

der

public class B // can access outer class members

{ int b;

void show()

{ cout << b;

cout << num; // cout << increment();

}

psvm (string args[])

{

A a = new A();

a.getNum(10);

b1.b = 50;

b1.show();

}

} // class A closed

Suppose

static public class B

{

/*

sout(num) X

can't access non-static members

sout(getincrement()); X

can access static members of outer class
only if create object of outer class

*/

/* in main()

B b1 = new B();

A.B b1 = new B();

*/

}

local inner class - Class inside the

// object of B class in method only not
public class A { method anywhere else

int num;

int increment(){

return num++;

}

void getnum(int n)

{

num=n;

public class B{

int b;

void show(){

sout(b);

}// show() ends

```
B b1 = new B(); b1.b=50;  
b1.show();  
} // void Num ends  
psvm(string args[]){  
A a1 = new A();  
a1.getNum(10);  
}  
} // class A ends
```

> anonymous inner class

```
interface A  
{  
public double int getA(int n);  
}  
class B  
{  
String s;  
void show(A a)  
{  
cout(a.getA());  
}  
}  
class C implements A  
{  
= public int getA(int n){  
return n+1;  
}
```

psvm(String args[]){

A a1 = new C();

B b1 = new B();

b1.show(a1);

}

create class during
anonymous class: create objects

psvm(String args[]){

A a1 = new A(){

public int getA(int n){

return n++;

};

Anonymous inner class

```
interface A
{
    public int getA(int n);
}

class B
{
    String s;
    void show(A a) {
        cout(a.getA());
    }
}

public static void main(String args[])
{
    A a1 = new A() {
        public int getA(int n) {
            return n++;
        }
    };
    // A a2 = new A(); not possible
    B b1 = new B();
    b1.show(a1);
}
```

```
psvm (String [] args){  
    B    b1 = new B();  
    b1.show (new A(){  
        public int getA(int n)  
        {  
            return n++;  
        }  
    });  
}
```

```
jButton1.addActionListener(  
    new java.awt.ActionEvent.ActionListener()  
    {  
        public void actionPerformed(ActionEvent e)  
        {  
            ==  
        }  
    } );
```

```
psvm  
{  
}  
b  
}  
}
```

```
}  
class  
{  
    pul
```

```
*psvm (String args[])
```

```
{ //a is abstract class
```

```
    B b1 = new B();
```

```
    b1.show( new A() {
```

```
        public int getA( int n )
```

```
        { return n++;
```

```
}
```

```
    } );
```

```
}
```

```
class B
```

```
{ public void show(
```

```
class X  
{  
    public void getX()  
    {  
        =  
    }  
    --
```

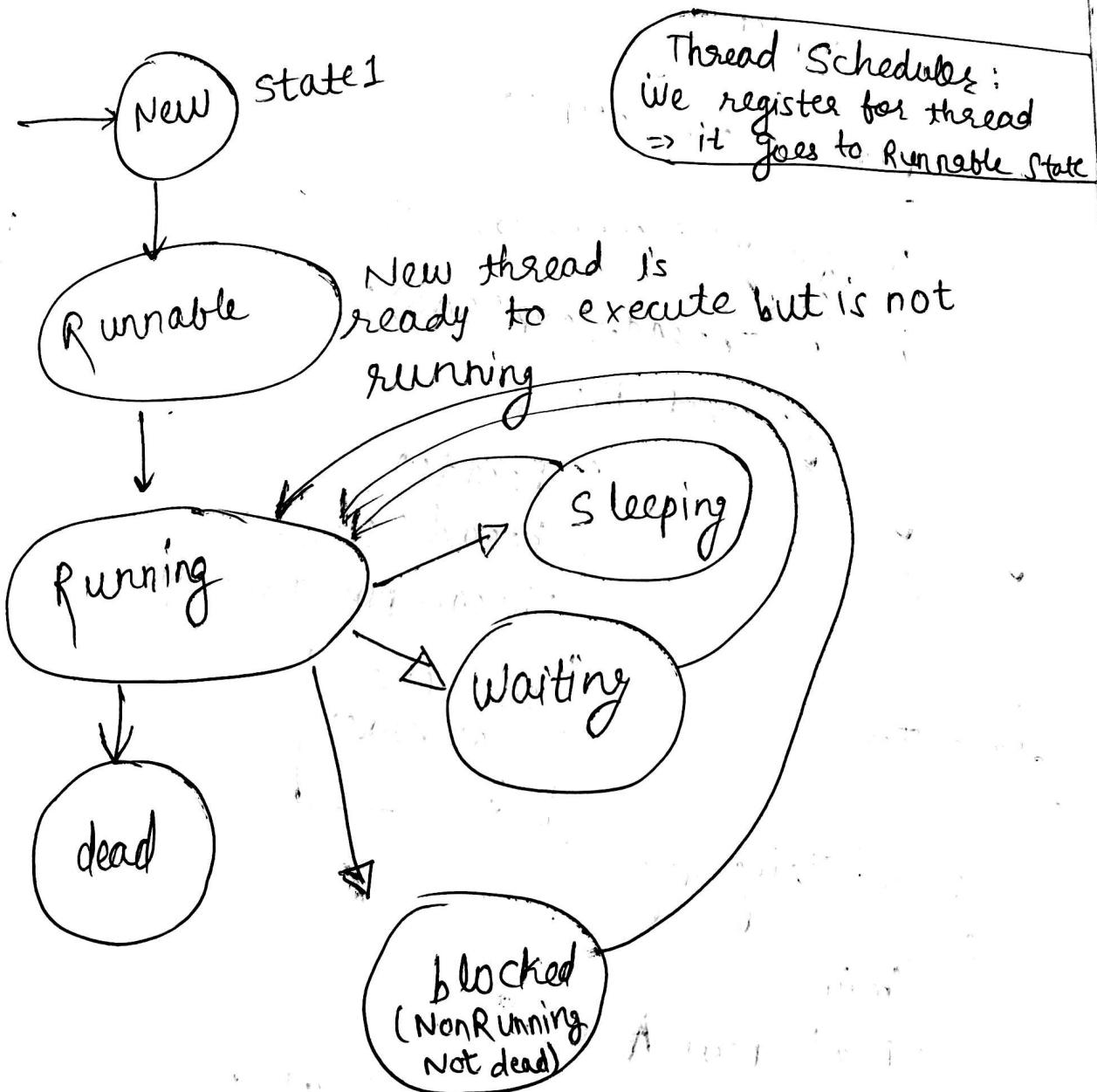
```
}  
X x1 = new X(){  
    public void getX()  
    {  
             //override  
    }  
};
```

```
X x2 = new X();
```

```
x1.getX(); }  
x2.getX(); } see the difference.
```

Dt: 14 Feb 2018

Thread: smallest part of the program that gets executed independently.
By default the whole program is single threaded. It is called main thread.



* Thread.sleep(200);

↓
ms - n

Runnable Interface
Runnable Method: contains only run()
method. ∵ needed to be
overridden.

Dt: 15 Feb 2018

class A implements Runnable

```
{  
    public void run()  
    {  
        for(int i=0; i<10; i++)  
        {  
            Thread.sleep(300);  
            sout(i);  
        }  
    }  
}
```

class Test

```
{  
    psvm(String args[])
```

//main

```
    A a1 = new A();
```

```
    Thread t1 = new Thread(a1);
```

```
    t1.start();
```

```
    Thread t2 = new Thread(new
```

```
        Runnable() {
```

```
            public void run()
```

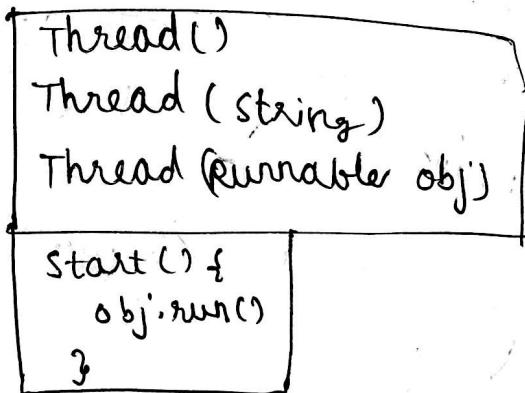
```
            for(int i=0; i<10; i++) {
```

```
                Thread.sleep(200);
```

```
                sout(i);
```

```
}
```

```
}; } t2.start(); t2.start();
```



We can
(Anonymous)

Runnable

{
}

};

Thread

new T

new T

{
}

})

psvm(String

//publ

A a



We can create object of Runnable in creating.
(anonymous classes).

```
Runnable r1 = new Runnable() {
```

```
    public void run() {  
        =
```

```
    };
```

```
Thread t2 = new Thread(r1);
```

```
newThread(r1).start();
```

```
new Thread(new Runnable())
```

```
{  
    public void run()  
    {  
        =  
    }  
}
```

```
}).start();
```

F
i
n
g
l

short
cut.

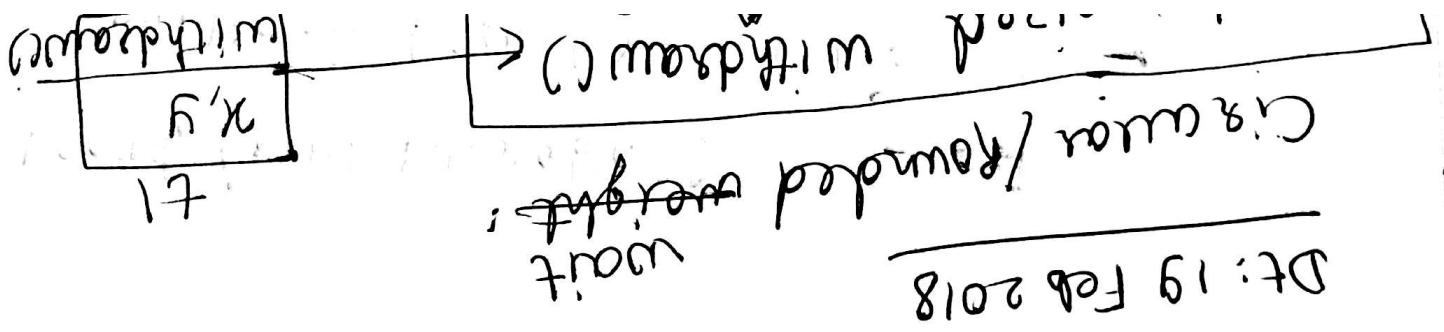


Fig 2 Repeat the above program by reading data from the file.
 providing standard behaviour.

Due to the file structure and its functionality it is trying to write new data on the same file by overwriting those already present in the file with an unhandled Java code

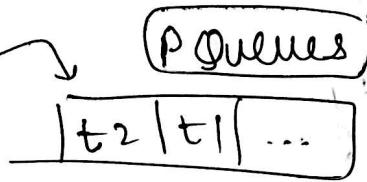
priority

Thread()
 Thread(string)
 Thread(Runnable, String).

new Thread(str);

t1.getName()
 t2.setName()

t1 200ms
 t2 200ms



main

Priorities 1 MIN_PRIORITY
 5 NORM_PRIORITY
 10 MAX_PRIORITY

t1.setPriority(3); → for min pri. Thread. MIN_PRIORITY

t2.setPriority(6);

#

int n=t2.setPriority();

Thread Scheduler

[t1 | main]

join()

class A

{

int sum;

run()

{

calcSum -

}

500ms

sum = 15

main()

{

t1.start();

t1.join();

in 500ms: ↓ (real)

sout(al.sum); 15.1
 joining t1 :: waiting
 for t1 to finish.

100

join()

join(milliseconds)

main()

{

 t1.start();
 t1.join(1000);

 cout << sum;

Dur
in b
blo
dea

On
su
ev

A
ai
K

Relation

Roll name student
No age %

RDBMS

query language

address

1

one
relation

i
Data

attribute

sql - structured
query language

DB2

Oracle

MySQL

MongoDB

DBMS software

Dt: 21 Feb 2018

execute()

~~exec~~
executeUpdate()

executeQuery() → Statement class

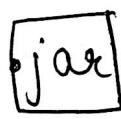
Statement st = con.createStatement();
↓
Connection object

~~DriverManager.getConnection()~~
Connection con = DriverManager.getConnection(
st.executeQuery();

staticMethods
↑

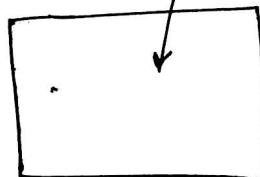
Java ↔ Database

jdbc drivers → group of classes
(.class files)



java archive
files

(bundle of classes).



DOWNLOAD  jar files

mysql> create database AWP;

mysql> use awp;

mysql> show tables;

mysql > describe log:

mysql> select * from log;

mysql> use awp;

~~mysql > create table student (roll no int(10) not null,~~

~~mysql > create table student (rollno int(10) not null,~~

→ name varchar(30),

→ rollno int(10),

→ per double(20),

→ primary key (rollno));

mysql > describe student;

Resultset rs = executeQuery(sql);

dt: 22/02/18

try {

stmt = com.createStatement(); → query
rs = stmt.executeQuery(); → to retrieve
data
executeUpdate(); →

not

at nu

Prepared Statement pst = com.prepareStatement(sql);

SQL Query: insert into Person (name, age, gender)
values (?, ?, ?);

pst.setString ("Amol", 1);

pst.setInt (25, 2);

pst.setString ("male", 3);

H/W: Repeat Person program using Prepared statement

Dt: 26 Feb 2018

glassfish\bin\asadmin.bat

Open glassfish (install) server in C: drive
inside that search folder glassfish

Again search

Open the folder

In side glassfish, search for bin.

Inside bin search for a file asadmin

Double-click on it to get server command window.

In the asAdmin command window write

start-domain domain1

↓
default domain

for glassfish serve

Open any browser, enter your application details

Command to stop server

stop-domain domain1.

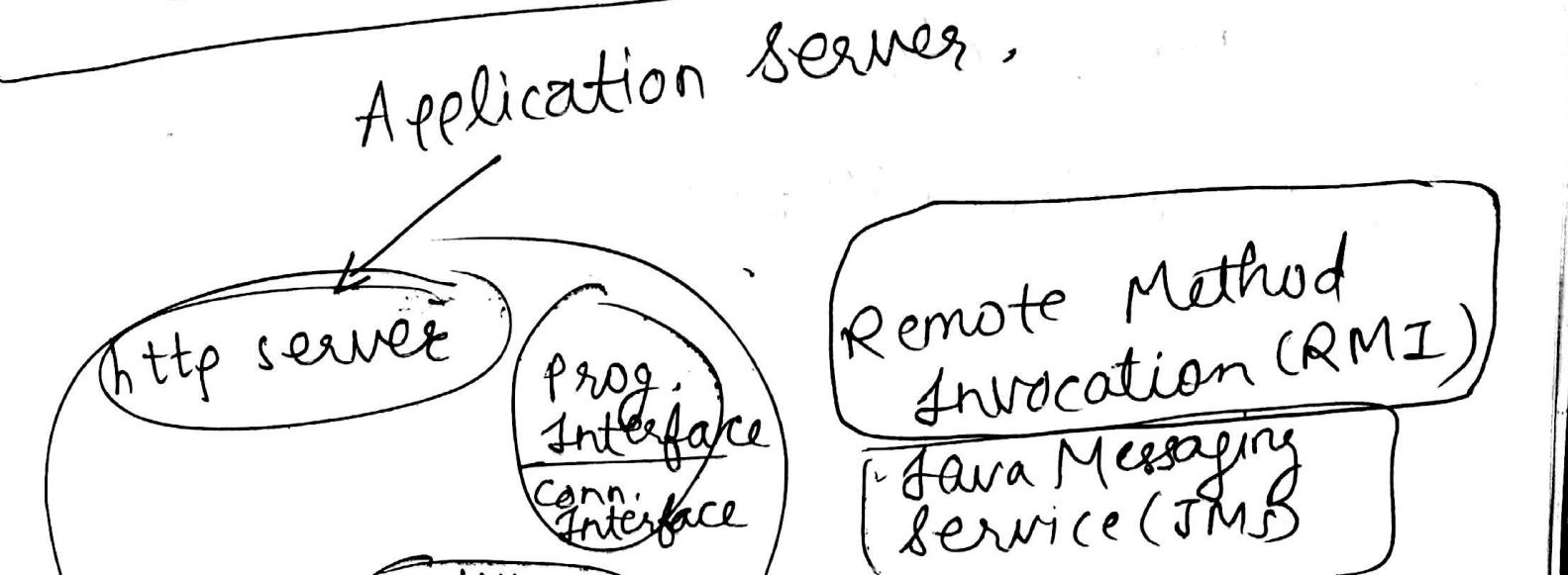
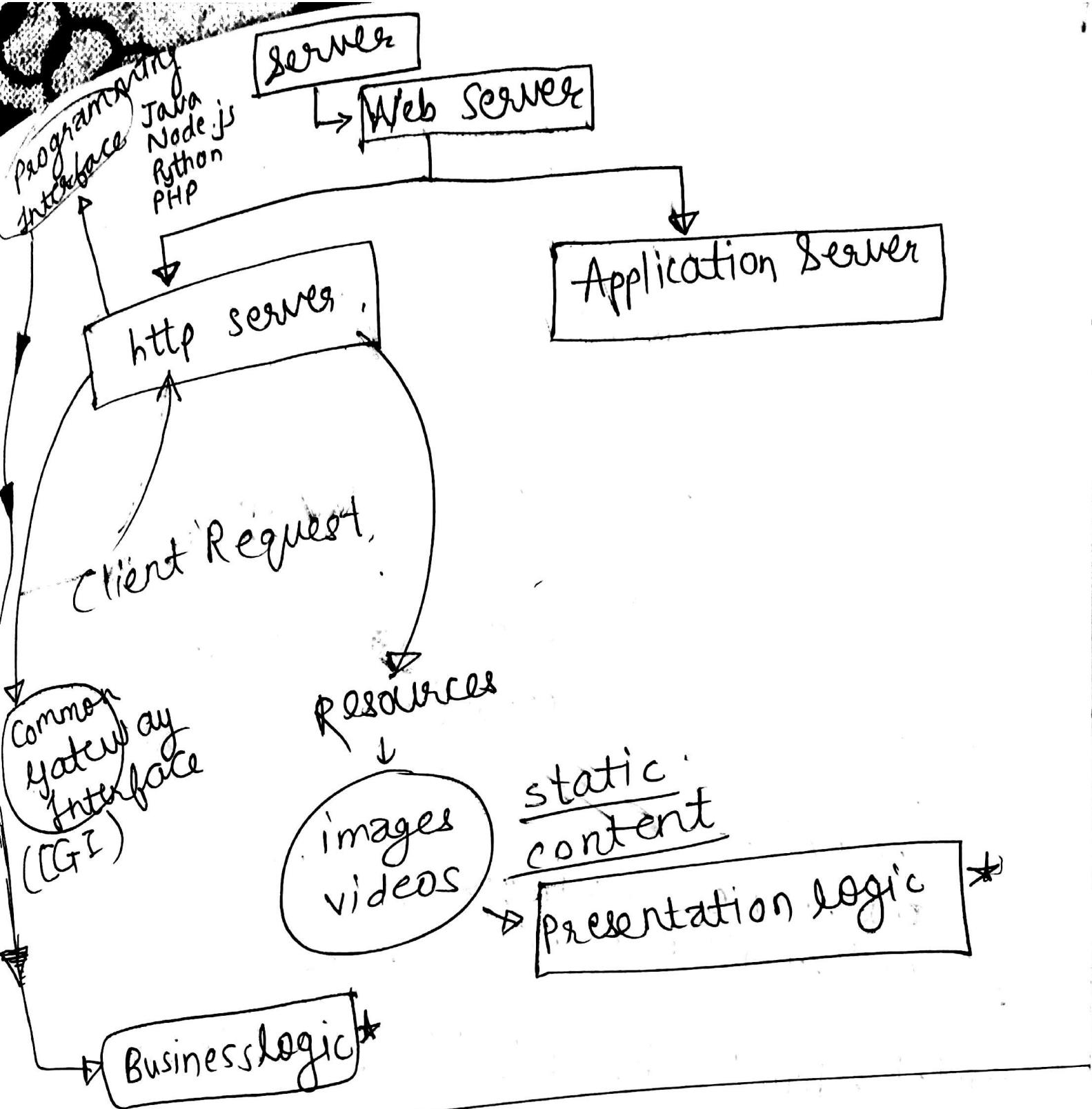
programming
interface
Ja
N
P
P

Client

Common
Gateway
Interface
(CGI)

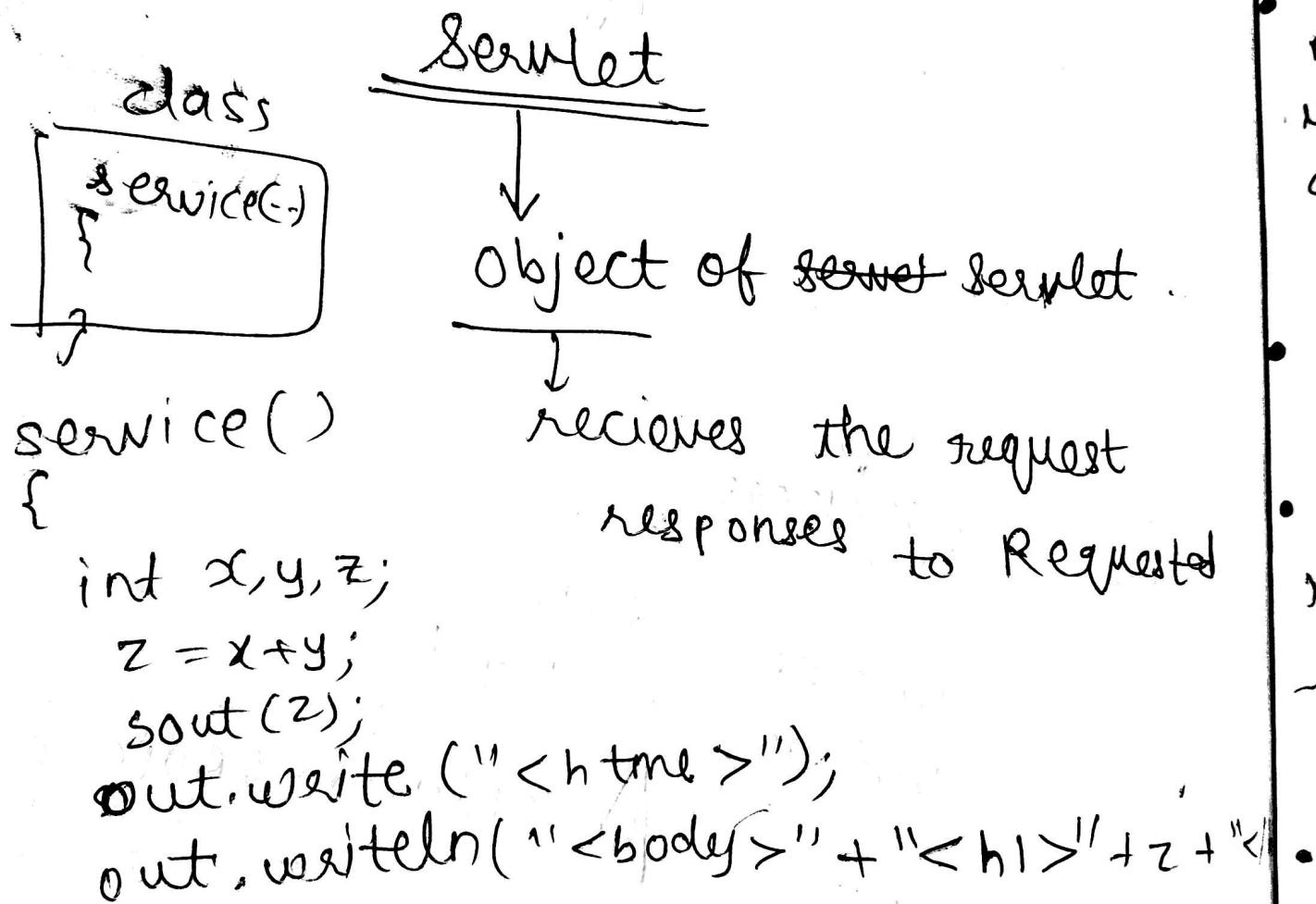
B

http



Application Servers

> glass fish
> Web logic (of Google) } }
 Java Application



2

Servlet: html code embedded inside a
java code.

↓
extends Servlet interface

* some points are very important:

- servlets are java objects which provides mechanism for processing the data as well as it provides the mechanism for displaying data

(Presentation logic)

- servlets are generally subclass of http servlet. (HttpServlet is a class).
- When servlets are requested by the user, after processing of the servlet, user can see only the contents but they cannot see the business logic.
- servlet uses objects of servlet context, servlet config, http session, request dispatcher, print writer, etc.

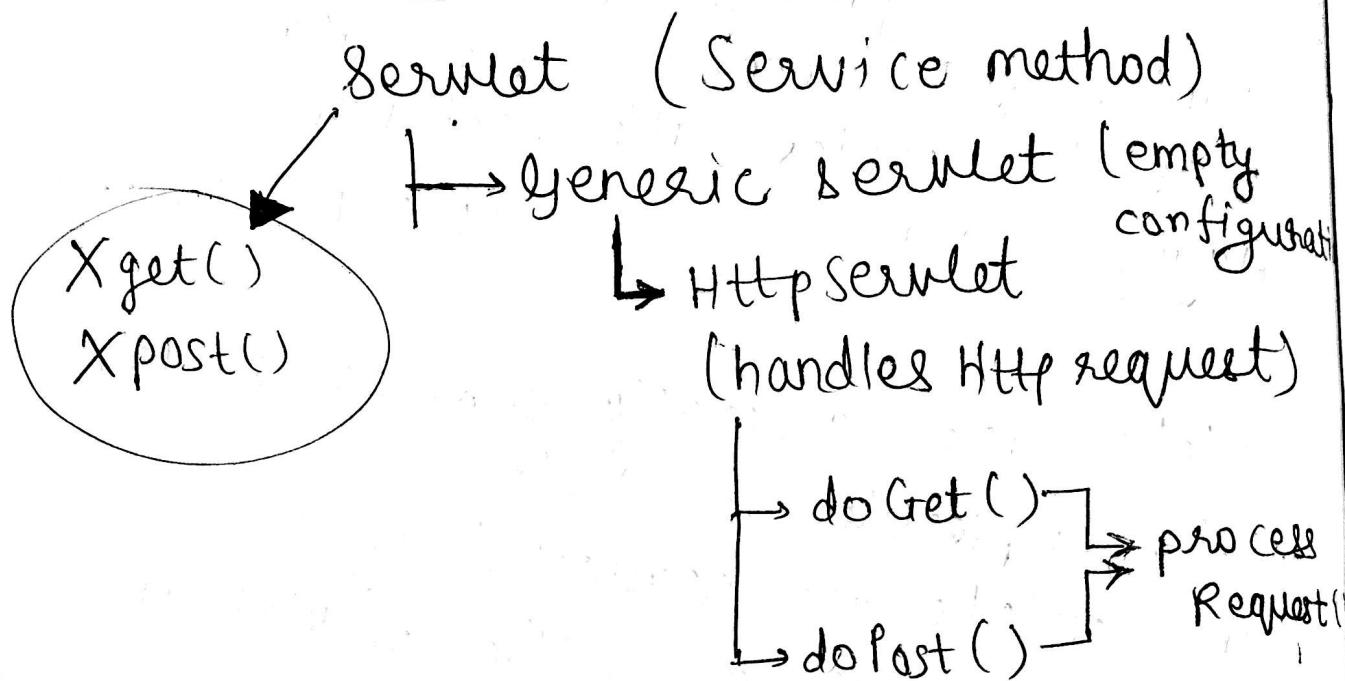
[Http Session]

- Writing alias name in URL pattern tag of web.xml is optional in jsp but it is mandatory in servlet.



- H/W: Read about the methods of the HttpServlet class
① doGet() ② doPost() ③ doHead()
④ doPut() ⑤ doDelete() ⑥ doOptions()

Dt: 28/02/18



We should be able to perform:

Ques. If user enters a state number
in textfield in index.html then
your response should display a
list of cities.

(wd)
(empty
config
queue)

Process
Request

dt: 01/03/18

http://localhost:9890/Servlet Test/Add?num1=10&num2=20

for parameters
↑ num1 = 10
↑ num2 = 20
↓ Servlet URL
↑ value of parameter
name of parameter

```
string param = request.getParameter("state");
<body>
<select name = "state" onchange = "calls(this.value)">
    <option value = "other"> other </option>
    <option value = "mh"> Maharashtra </option>
</select>
<script type = "text/javascript">
    function calls(v1){
        if(v1 == "mh"){
            document.location.href
            = "http://localhost:9890/Servlet Test/Add?state=mh";
        }
    }
</script>
</body>
```

```
out.println("<body>");  
out.println("<ul>");  
out.println  
while(sc.hasNext()) {  
    out.println("<li>");  
    out.println(sc.nextLine());  
    out.println("</li>");  
}  
out.println("</ul>");  
out.println("</body>");  
out.println("</html>");
```

<h3> Download this </h3>

-> <div>
 <a href = "Downloadfile?param=vid"
 <h2> video </h2>

</div>

import org.apache.commons.io.FileUtils;
import org.apache.commons.io.IOUtils;

import org.apache.commons.IO.jar

✓

respon
strine
File
if ()
}]

```
response.setContentType("application/octet-stream");
String attr = request.getParameter("param");
File srcFile=null;
if(attr.equals("pvideo")){
    response.setHeader("Content-Disposition",
"filename = \\""+fName+"\\\"");
    srcFile = new File(path1+fName1);
}
if(attr.equals("pmp3")){
    response.setHeader("Content-Type",
"audio/mpeg");
}
if (attr.equals("videoplimage")){
}
InputStream in = fileUtil.openInputStream(srcFile);
IOUtil.copy(in, response.getOutputStream());
```

~~servlet config~~ = globalvar

ServletConfig config = getServletConfig();
int id = Integer.parseInt(config.getInitParameter());
sout(id);

Scope:
Servlet
(not context)

* scope: Context → Access this in the following way:

ServletContext context = getServletContext();
String myname = context.getInitParameter("my name");
sout(myname);

To redirect a request to another resource:

response.sendRedirect("http://www.google.com");
~~response.sendRedirect(request.getContextPath() + "/index.html");~~

init,
context

+ "/index.html");

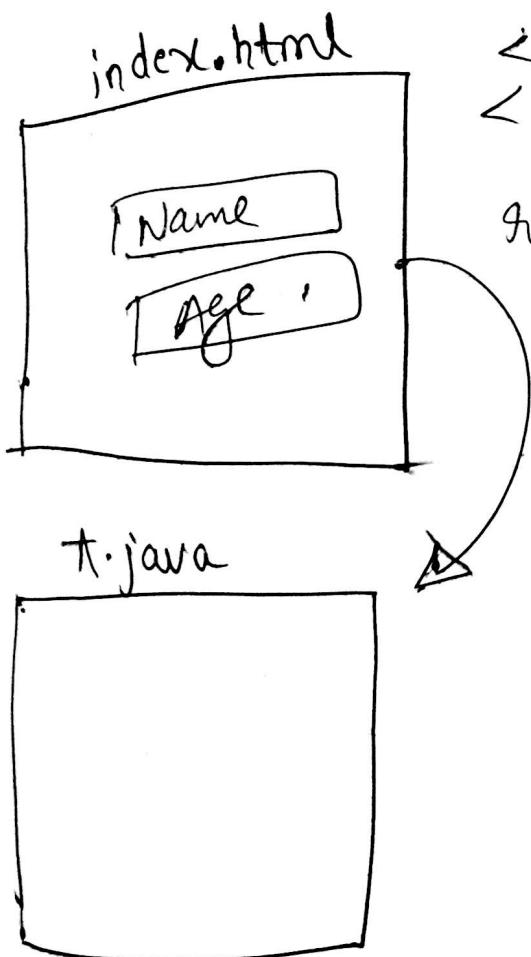
Dt: 05/03/18

web.xml

```
<context-param>  
</context-param>  
<context-param>  
</context-param>  
<servlet>  
<init-param>  
</init-param>  
</servlet>
```

SEARCH for
index.html
OR
index.htm
OR
index.jsp

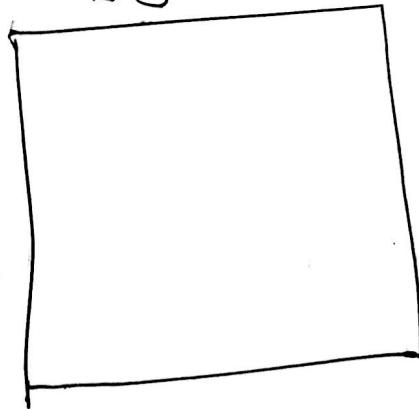
in
web page folder



<name="name"/>
<--. name="age"/>

request
(name, age)

B.java

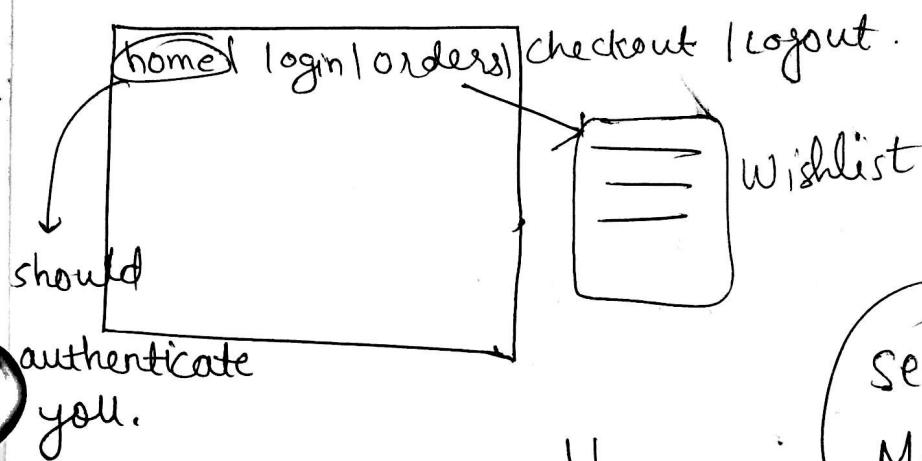


forward method to
send same request to
another servlet
(html page also is
possible).

Request Dispatcher rd = request.getRequestDispatcher("myHtml.html");

rd.forward(request, response);

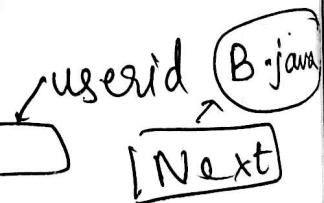
home.html



10 - 11
1 hr

Session Management

manage user credentials



- 1) hidden form field.
- 2) Url Rewriting
- 3) Http Session
- 4) Cookies

Url Rewrite

La web

request
user

Session

home log

Sessio

T

H
S
C

URL Key
`Home `

request

user

myHtml.jsp

```
<% String user=request.getParameter("user"); %>
```

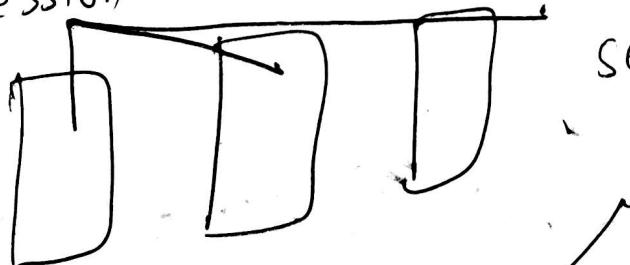
session object

home | login | orders | checkout | logout

Session obj

Bind Parameters.

session



HttpSession session;
session = request.getSession();

session.setAttribute("user", str);

attribute name
value

attribute (object)

```
HttpSession session;  
session = request.getSession();
```

```
String uid=(String)session.getAttribute("user");
```

```
String uid=(String)session.getAttribute("user");
```

Noting the H/W on dt: 05th March 2018

- Ques. Create a simple application with home / login / orders / checkOut / logout tabs and manage the sessions using
- ① hidden form field
 - ② url rewriting.

Form action = "receivehidden" method = "post"
<input type = "hidden" name = "id" value = "tid + >"

Solve / Complete your H/W on time.

```
out.println("<h2><a href = \"MyHomePage?  
username = " + username + "\">Home  
+ "<a href = \"userlogin.html\">Login</a>  
+ "<a href = \"#\">Orders</h2> <br><br>
```

for expiring session
session, invalidate()

session = request.getSession()

Session.setAttribute("user", "Suresh")
Session.getAttribute("user")

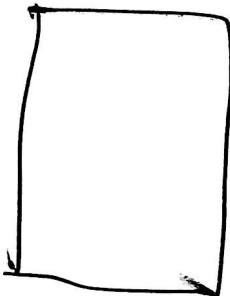
Session. setAttribute("id", 1)
Session. setAttribute("name", "Suresh")

Session. setAttribute("age", 26)
Session. setAttribute("page", "index")

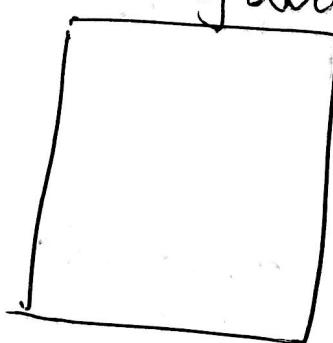
Dt: 08/03/18

Session Management

A.java



B.java



HttpSession session;

session = request.getSession();
getSession(true or false);
getSession(false) → return

existing session if it exists

else ~~not~~ NULL :

session.setAttribute("user", "value");
/ ,

setInterval(time(seconds)) key

Cookies → java.

A.java

```
Cookie cookie;  
cookie = new Cookie("userid", "bharat");
```

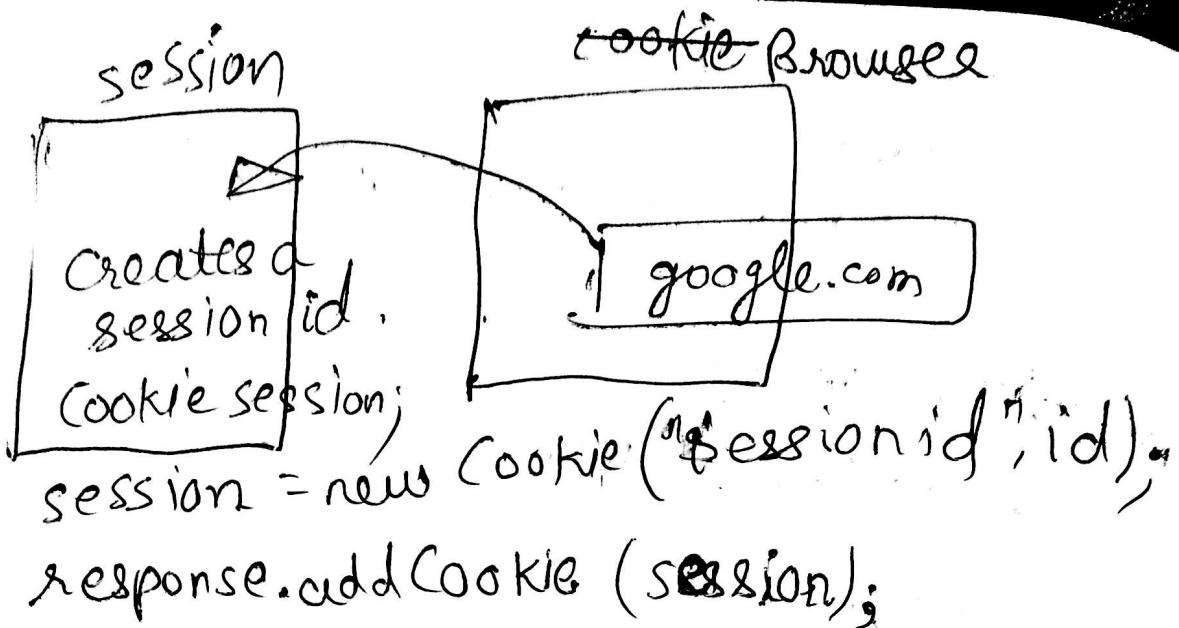
~~response.setCookie~~

```
response.addCookie(cookie);
```

B.java

```
Cookie ck[];  
ck = request.getCookies();  
for(Cookie temp : ck)  
{  
    sout(temp.getValue());  
    if (temp.getValue().equals("bharat"))  
                
                
    }
```

getname - userid



What Browser does?

```

    :Cookie ck[] = request.getCookies();
    String value;
    for each (Cookie temp: ck) {
    }
    ck[0].value = ck[0].getValue();
  
```

When cookies are disabled, URL
Rewriting is used by server.

Servlet

request
response
session
cookies
context
config.

JSP

Objects are already created

Java Servlet Pages

<html>
<head>—
<body>

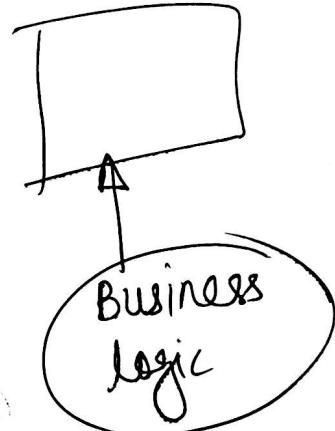
JSP - embedding java in html.

</head>

<% %>
Servlet

<%—%>
—
<body>
</html>

Presentation logic



jsp

3 types of tags

<h2>

<% = %>
</h2>

%>

① Expression tag

② Scriptlet tag

③ Declaration tag

<h2> Welcome <% = str %>

<head>

<% → scriptlet tag

String str = request.getParameter("username")

~~&~~ %>

<%= str %>