Advance Java

Types of Application ----->

1. Standalone application
   a. The application which is used without internet is called as standalone application.
   b. This application present at client side.
   c. To develop standalone application programmers using core Java.
2. Web application
   a. The application which is used with internet & browser it is called as web application.
   b. This kind of application present at client side & server side.
   c. To develop web application we have to use Advance Java.

Note: The frontend of application present at client side & backend of application present at server side.

3. Mobile application
   a. The application which must be installed in mobile phones & user can used them with or without internet after installation it is called mobile application.
   b. This application present at client as well as server side. Hence, it is called as client server application.
   c. To develop mobile application programmers using android .

Server ---->

1. Server is a mediator between users & programmers/developers for sharing the application.
2. Programmer using server to store the application & users are using server to access the application.
3. Users will communicate with server is referred as Request whereas server procced the request and send response to the user.

Features of Application -

There are three important features of application.

1. Loose coupling -

   Adding a new feature to the application which does not affect old features of the application is referred as "Loose coupling".

   Tight coupling ---> It is the opposite behaviour of loose coupling. Adding the new feature to the application which effects the existing feature in the application is referred as "Tight coupling".

2. Multithreaded -

An application which can be used by multiple users at the same time is referred as "Multithreaded".

Single Threaded ---> It is the opposite behaviour of multithreaded. An application which is used by only one user where other users have to wait is referred as "single threaded".

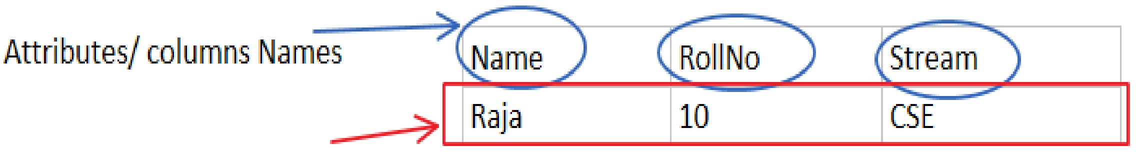
3. Session -

The time interval given for the users by the programmers to complete a particular task is referred as "sessions".

Note: - Single-threaded & tight coupling are worst behaviours/ features of an application whereas multithreaded & loose coupling are best behaviours.

DATABASE

1. As a programmer we are using database to store user information permanently.
2. To working with advance java we are using MY-SQL database and it stores user information in the form of tables

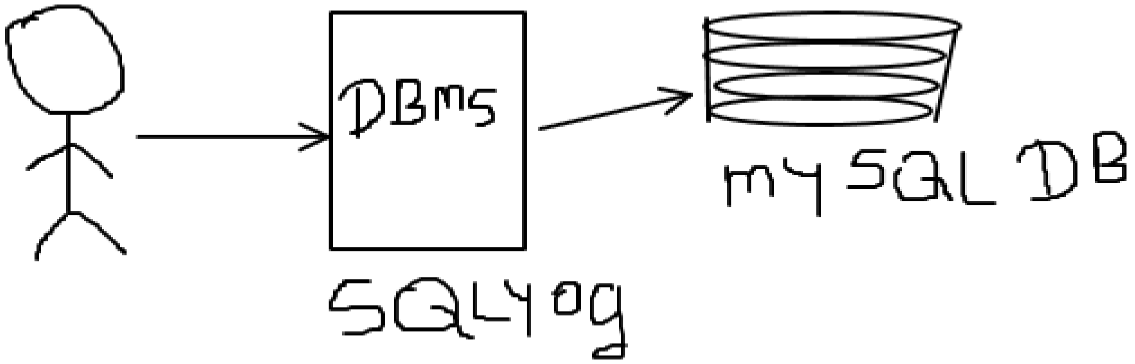

Attributes/ columns Names

Name	RollNo	Stream
Raja	10	CSE

Student-→ table name

3. Programmers cant access user information directly from database, So we have to use DBMS(Database Management System) to access user info from DB.
4. In MYSQL DB user information is stored in the form table. Hence, use RDMS(Relational Database Management System).
5. To get RDBMS in system & to work with MYSQL we have to install SQLYOG application.

6. To perform operations on the Database by using DBMS programmers have to use SQL (Structured Query Language). Operations like insert, delete, update & retrieve.



Steps to be followed in eclipse to develop web application.

1. Create a project & name the project.
2. Create a package within 'src' folder with given standards below.

Org/com Ex. com

Company Name Ex. google

Application name/ project name Ex.Gmail

The package name for the above example is -
com.google.gmail

Note : All the package names must be in a lowercase.

3. Create a class along with main method.

Generate OTP

Requirement - OTP ----> Random in nature, integer, 4/6 digit.

Note: for loop is used for sequential manner.

1. Random class is used for Random manner.
2. It is inbuilt class which is present in java.util package & It is used to generate random numbers.
3. All the reference variable names must be same as class name.
4. nextInt() is an polymorphic method which is working with Random class as well as Scanner class.
5. If nextInt() is called by using Random class reference variable then it is used to generate random numbers.
6. If nextInt() is called by using Scanner class reference variable then it is used to get integer values from user.
7. System.err.println() is used represent error & invalid message. It is appears in red colour.

Generate 6 Digit OTP

Jar Files

1. Jar stands for java archive.
2. Compressed form of files & folders is referred as zip file.
3. Compressed form of java programs/projects is referred as jar files.

4. As a programmer we use jar files for sharing the program from 1 developer to other developer in company. Hence, it is used as communication medium between developers.
5. The Steps to be followed to cover java project/program into jar files are
    a. Select the project then right click on it.
    b. Select export option
    c. Expand java folder present in export wizard.
    d. Select jar file option then click on next.
    e. Browse the appropriate location to create jar files.
    f. Name the jar file and click on save then click on finish.

Java Build Path

1. All java application are executed because byte code in case byte code is deleted, programmer have recompile code.
2. The contains of jar file is similar to the contains of a program where each program consist of two files.
    a. .java file(source code)
    b. .class file (byte code)
3. Giving the permission for the external files into project is referred as "Java Build Path".
4. The steps to be followed to perform Java Build Path are as follows:
    a. Select the project
    b. Right click on project & select properties option.
    c. Select java build path option & click on Libraries.
    d. Click on "Add jar files " option.
    e. Select appropriate project to which the external file was copy-pasted.
    f. Select external file of project & click on "Apply & close".
5. As a programmer we can execute jar files because it contains .class files.
   To execute jar files programmer have to right click on jar files and then choose "Run as" option & then select "Java application".
6. To check the source code which is present in jar files programmers have to follow below steps.
    a. Go to project explorer.
    b. Open Java Library / Referenced libraries.
    c. Select any jar file and open it.
    d. Open the package & decompile the .class file.
7. The process of converting byte code into source code is referred as De-compilation. In eclipse we used "fernflower" de-compiler.
8. All inbuilt classes is given with a byte code. As a programmer we checked the source code of inbuilt classes by using de-compilation process.

Note: Java library contains all the inbuilt classes whereas Referenced library contains external jar files classes i.e. User-Defined.

API

1. API stands for Application Programming Interface.
2. As a programmer we use API for Inter Application Communication.
3. Sharing user information from one application to another application is technically referred as Inter application communication. (Purpose of API)
4. API is a terminology which is used in javaEE perspective.
5. The contains of API are classes & interfaces where classes are differentiated as Helper class & implementation class.

Java perspective	Java EE perspective
Super class	Helper class
Sub class	Implementation class
Interface	Interface
Packages	API

6. The class which contains common properties is referred as Helper class.
7. The class which contains specific properties is referred as Implementation class.
8. Collection of non-static abstract method & static final variables is called as Interface.
9. The Combination of Helper class , Implementation class & interface is called as Application Programming Interface.(Definition)

Note: All the packages of Java perspective corresponds to API's of JavaEE perspective.
for example:

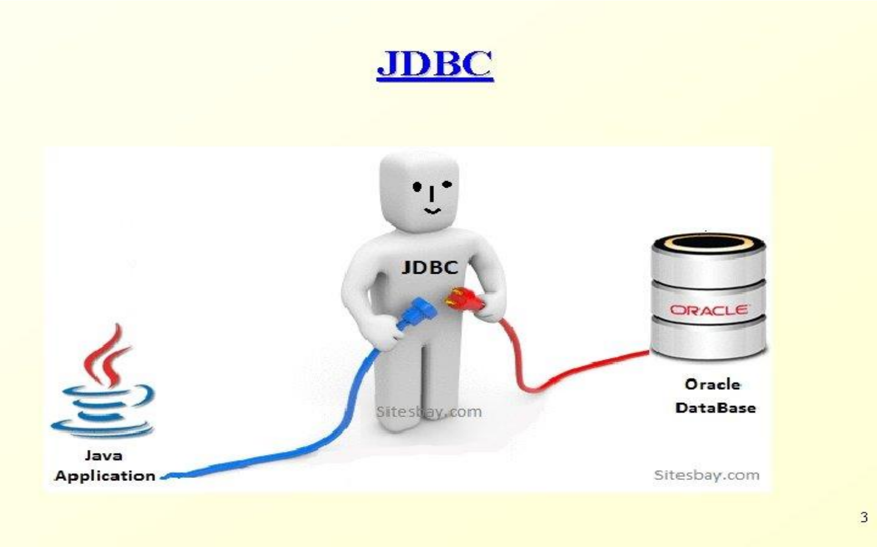
java.lang --------> java API
java.util ---------> utility API

Drivers

1. Drivers are also referred as Translators.
2. The process of converting one language to another language is referred as Translation.
3. For example-
   a. Compiler is a driver because it converts source code into byte code i.e. High level language into Numerical language.
   b. JVM(Interpreter) is a driver which converts byte code into binary code i.e. Numerical language into Binary level language.

JDBC Architecture

1. JDBC stands for java database connectivity.
2. To connect Java application with Database application programmers using JDBC.
3. To store user information inside the database.



Note: As an user information is given to the application but programmers not store this information into application instead of they are storing it into Database.

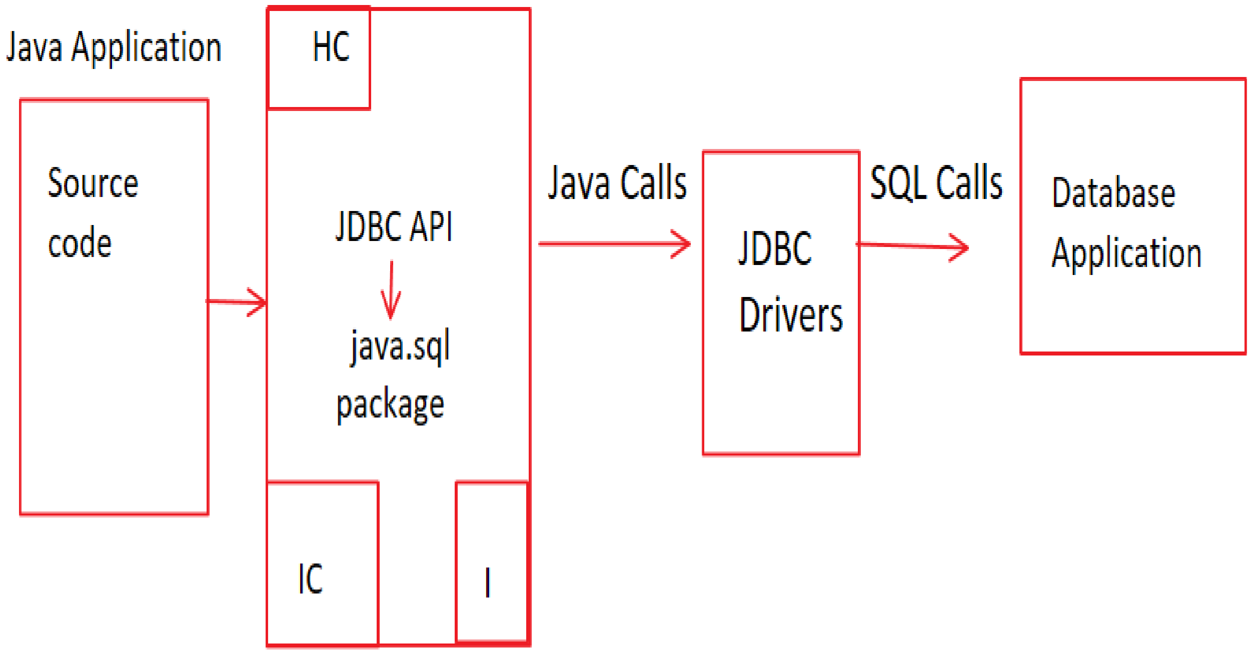
4. To understand process of JDBC programmers using JDBC architecture which consists four important components.
   a. Java application
   b. JDBC API
   c. JDBC Drivers
   d. Mysql Database
5. Java application consist of source code which is responsible to take information from user into application.
6. JDBC API helps programmer in sharing the user information from java application to database application.
   Note: JDBC API corresponds to java.sql package.

7. JDBC drivers translates java calls into sql calls which can be understood by database application.
8. Database application is used to store user information in the form of tables.
9. As a programmer we cannot access database directly. Instead we use the concept called it as DBMS (Database Management System).

Note: As a programmer we are using MYSQL database application.

10. To perform operations on the data present inside database, as a programmer we have to use SQL (Structured Query Language).
11. The software to be installed for the working with Mysql Database. a) MYSQL Database server

b. SQLYog application.



Properties File:

1. As a programmer we can create many types of files based on the requirements.
   a. General Purpose file - it is used to represent contents in the form of paragraphs or text
   b. Properties File - it is used to represent contents in the form of key-value pair.
   c. Serializable file - it is used to store object inside file.

2. To work with properties file programmer will make use of FileReader class which is present in java.io package. It is used to read file & throw the checked exception called it as FileNotFoundException.
3. Representation of key- value pair is given by Properties class, which present in java.util package.
4. To use the contents of properties file in java class, programmers will make use of load().
5. getProperty() is a Non-static method and overloaded method which is basically used to retrieve the value based on the specified key as an argument.
6. If Specified key is not present in the properties then it returns null value without throwing any exception.
7. The method signature for get property method is "+ String getProperty(String key)". This method returns value if key is exists in the file. If key is not exists in the file then it returns null.
8. "+ String getProperty(String key, String msg)" this method returns value if key is existing in the file. If the key is not present then returns the message which is given as second argument.
9. For example:

```
user = root
Password = 12345
```

Properties file

Note:

1. If file is not present at specified path which mentioned for FileReader then it is throwing FileNotFoundEXception (checked exception).
2. If input & output gets interrupted i.e. not present in proper manner then load() is throwing IOException(checked Exception)
3. Throwable class contains printstacktrace() which is non-static method & responsible to return detail information about exception like name, line number, reason, etc.

Working with SQLYog

1. SQLYog application works as an intermediate between programmer & MYSQL database.

2. To work with sqlyog application, programmers have to connect to the mysql database by providing the appropriate password and then click on connect option.

3. To create the database follow the below steps.
   a. Go to root@localhost option present at left side of sqlyog.
   b. Right click on this option then select create Database option.
   c. Give the name to the Database and click on create.

4. To create the table in the database follow the below steps.
   a. Right click on the Database ----> select option create table.
   b. Provide column names as field names with the appropriate datatype, length & constraints.
   c. After providing all column information, click on create table option.
   d. Finally name the table and click on OK.

   Note: pk ---> primary key which used to allows unique values in specified column.
   not null is used to does not allows null values in specified column.

5. After creating the table programmers can see the table contents by using "view data" option.
   To get this right click on the table name.

6. To insert data inside the table programmers have to write insert query in Query section by using following syntax.
   Syntax --->

   insert into table_name values(column1,column2,column3)

   Create table to store userinformation like name, mail-id, contact and insert data in

7. To update data of table use update query as per following syntax:

   Syntax ---> update table-name set column1 = value1 where <condition>

   Example --> Update student set studentName = 'xyz' where studentName = 'Rani'

8. To delete data from table programmers have to make use of following query:

   Syntax --->delete from table-name where <condition>

   Example ---> delete student where rollNo=101

Create table to store Debit card information & perform insert, update operation.

Uniform Resource Locator.

1. Url stands for uniform resource locator
2. Every database will be given with four important information
   a. Protocol
   b. Host information
   c. Port number
   d. User information
3. Protocol is the set of rules which must be followed while using the application.
4. Host information is the way of access the application. We have two types of Host information.
   a. Local host
   b. Remote host

5. Accessing the application which is present in the server & server is present in some remote location. This way of accessing the application is technically referred as remote host.
6. Accessing the application which is present at client machine is called as localhost.
7. Port number is a gateway through which data flows.
8. User information is used to provide security for the data by giving username & password.
9. "Combination of protocol, host info, port number & user information is referred as URL"

   Note: The contents of Database is user information whereas the contents of server is application. Hence the protocol for the database & server is different.
   For example :
   1. The server protocol is "https"
   2. The protocol for the database is --->

   "jdbc:subprotocol" where sub-protocol is name of database.

10. Syntax of URL is

    Protocol://host info:portNumber?userInformation

11. For example:

    a. If we use MYSQL DB then protocol would be jdbc:mysql & we are working for localhost information with 3306 as portnumber & user-information as user-name is root and password is 12345.

    jdbc:mysql://localhost:3306?user=root&password=12345

    b. If we use Oracle DB then protocol would be jdbc:oracle & we are working for localhost information with 1521 as portnumber & userinformation as username is scott and password is tiger.

    jdbc:oracle://localhost:1521?user=scott&password=tiger

Note: URL plays important role to establish connection between java application & database application.

Steps Of JDBC

1. As a programmer we have two approaches to develop the application.
   a. Without using inbuilt libraries.
   b. With using inbuilt libraries.
2. If an application is develop without inbuilt libraries then time required for developing the application would be more. Hence, it is not the suitable approach to develop the application.
3. If an application is develop using inbuilt libraries then time required for developing the application would be less. Hence, this is the best approach to develop the application.

Note: To write JDBC programs, programmers will make use of inbuilt library by name java.sql package.

There are five steps of JDBC:

1. Establish the connection between java application and database application.
2. Create a platform.
3. Execution of SQL Query.
4. Process the resultant data.(optional)
5. Close the connection.

Step 1 - Establish the connection

1. Inbuilt method which is used to establish the connection is getConnection().
2. Get connection method is on static method present in DriverManager class / java.sql.DriverManager.
3. getConnection() is an static & overloaded method with three variations.
   a. + static Connection getConnection(String url)
   b. + static Connection getConnection(String url, Properties propt)
   c. + static Connection getConnection(String url, String user, String password)
4. getConnection() takes URL as an argument, which consist of the information about Database to be connected.
5. getConnection() throws SQLException which is checked exception and it is handle by try & catch block.
6. The return type of getConnetion() method is Connection Interface.
7. Syntax:

```
Connection connection = DriverManager.getConnection(Sting url);

Interface      reference   Helper class        static & overloaded method
Present in     variable     present in          for establishing connection
Java.sql        name       java.sql
Package
```

Steps 2 - creating a platform.

1. JDBC programs consist of both java code and SQL queries. Java compiler can understand the syntax java code only.
2. SQL queries which is present in the JDBC program can be understood only within the database.
3. To make SQL queries reach the database programmer will be creating platform.

Note: The main purpose of platform is to carry SQL queries from Java class to Database.

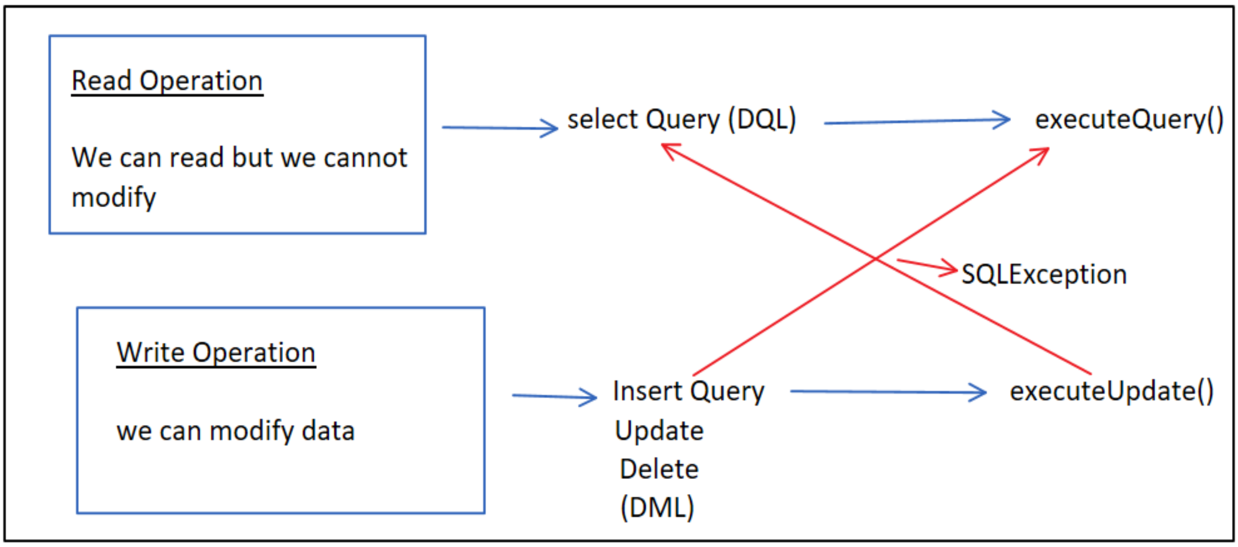
1. There are three types of platform
   a. Statement type
   b. PrepareStatement type
   c. CallableStatement type
2. If query contains hardcoded values then programmers have to prefer statement type platform.
3. If query contains runtime values then programmers have to prefer PrepareStatement type platform.
4. If query contains both hardcoded and runtime values then programmers have to prefer CallabaleStatement type platform.
5. Syntax for Statement Type -

```
Statement   statement   =   connection.createStatement();
```

Interface Present in Java.sql Package	reference variable	reference variable of Connection Interface	Non-static method to create Statement type platform

Step 3 :- Execution of SQLQueries

1. In this step we have two operation performed by programmer.
   a. Write operation
   b. Read operation
2. The operation which can modify the content of the table is referred as write operation. Programmatically write operation is represent by DML i.e. insert, update & delete query.
3. To perform write operation on table programmers using executeUpdate().
4. The operation which cannot modify the content of the table, but we get the data is referred as read operation. Programmatically read operation is represent by DQL i.e. select query.
5. To perform read operation on table programmers using executeQuery().

- 

Read Operation

We can read but we cannot modify

select Query (DQL)

executeQuery()

SQLException

Write Operation

we can modify data

Insert Query
Update
Delete
(DML)

executeUpdate()

Note: If there is mismatch between the type of operation & method used for its execution then it results in SQLException.

Assignment -

Create Car table with car no. as primary key column, car color & car brand as columns.

- Using the above table, WAP to add three records into the table.
- Using the above table, WAP to change the car color based on car no.
- Using the above table, WAP to remove one record from table based on car no.

Step4 : Process the Resultant data

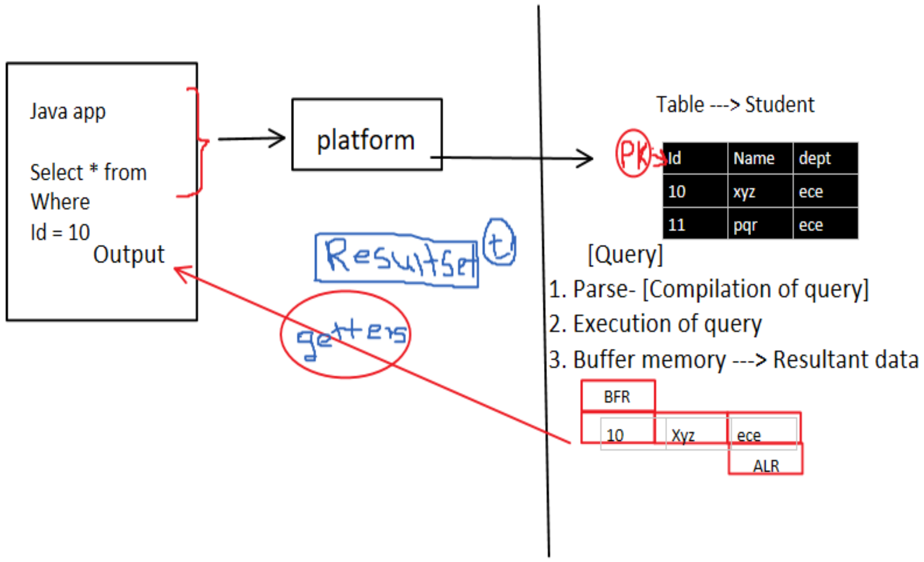
1. As a programmer we perform encapsulation process to avoid invalid values which are pa members.
2. The main purpose of setters method is, to pass the valid values for data members.
3. The purpose of Getters method is, to return/retrieve the values from data members.

Note: Programmer can make use of setters & getters in any program whether the program is encapsulation process or not.

1. The data which is present in the table is referred as actual data.
2. Once query reaches Database, It performs three important functions.
   a. Parse - Compilation of Queries
   b. Execution of Queries
   c. Buffer Memory
3. The output of read operation (select), after query execution is stored in buffer memory resultant data.

Note: To differentiate actual & resultant data in Database assigns BFR & ALR for resultant data
where, BFR stands for Before first recored and ALR stands for After last record .

4. The output of write operation(Insert, Update, Delete) is only to the Database table. Hence buffer memory contents will be empty.
5. To retrieve resultant Data from Buffer memory to the Java class programmers will make use Getters, which is the part of ResultSet Interface.
6. By Default, Database points to BFR position. To move the cursor from one position to another position, we have to make use of next().



Id	Name	dept
10	xyz	ece
11	pqr	ece

[Query]
1. Parse- [Compilation of query]
2. Execution of query
3. Buffer memory ---> Resultant data

BFR		
10	Xyz	ece
		ALR

Assignment

a. Create employee table with columns empId ----> primary key , empName, salary
b. Insert the data in the employee table.
c. Display all records from employee table whose salary is between 3000 to 5000.
d. ✏️ Display all records of employee whose empId is an prime number.
e. ✏️ Display all records of employee whose empId is even number.

Salary > 3000 and salary < 5000

Prepared Statement

a. As a programmer we prefer PrepareStatement platform to pass runtime values, which results in flexibility in the code.
b. To pass the runtime values, programmers will make use of placeholder concept. It is denoted by '?'.
c. To take runtime value from user, we have to make use of Scanner class.
d. To pass those runtime values to the placeholder, programmers will make use of setters.
e. Setters method consist of two arguments.

    a. Position of the placeholder to which runtime values has to be pass.
    b. Value to be pass.

f. We have to pass the values for placeholder before step. No. 3

g. The inbuilt method which is used to create PreparedStatement platform is preapareStatement(String query).

h. Syntax:

    PreparedStatement preparedStatement = connection.prepareStatement(query);

Note: While working for statement type platform, we have to pass the query in step no. 3 , which indicates that program consists of hardcoded values.

While working PreparedStatement type platform , we have to pass query in step2 of JDBC which indicates that program consists of runtime values.

Stored Procedures

a. The methods present in sql which are used for a storing a queries.

b. If we write the query within java application in order to compile & execute the query involves long process which in effect the performs of application.

c. If we write the query in stored procedure within a database application in order to compile & execute the query takes less time & increases the performance of application.

d. Steps to create a stored procedure.

    - Right click on database then select option create stored procedures.
    - Provide appropriate name for stored procedure and click on create.
    - Write Query in between the keywords 'Begin' and 'End'.
    - In order to compile the written query click on execute all queries option.

    In order to execute the query that is present within a stored procedure in DB we need to call the stored procedure from java application, to perform this operation programmers using CallableStatement platform.

CallableStatement Platform

It is an interface present java.sql package.

As a programmer we make use of a callableStatement platfrom to call stored procedure.

Syntax - (calling stored procedure)
{call database.storedProcedureName}

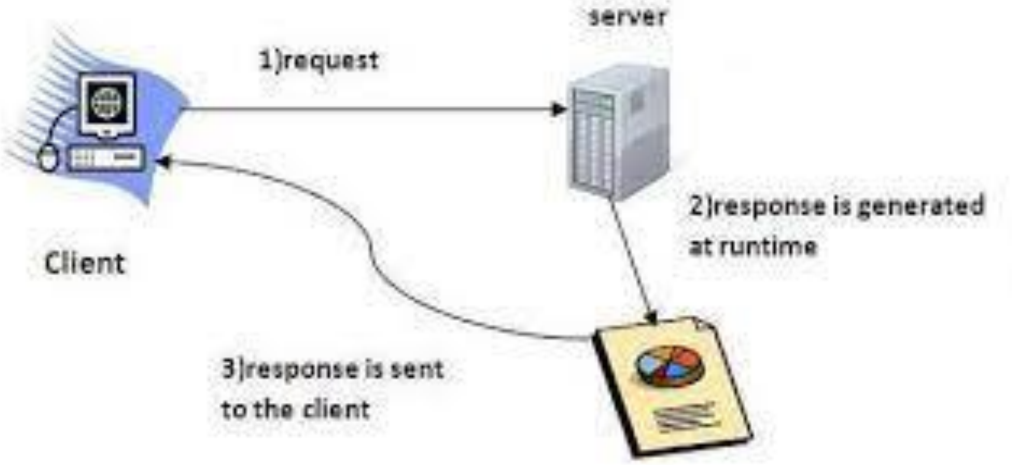
To create CallableStatement platform programmers using prepareCall() method and argument it accepts call stored procedure.

Syntax

```
CallableStatement callableStatement = connection.prepareCall(call);
```

Servlet

1. Servlet is used to develop the web application.
2. To provide the web application to the user programmers going to use server.
3. Server is mediator between programmer & user which is responsible to accept the request from user and give the response to the user.

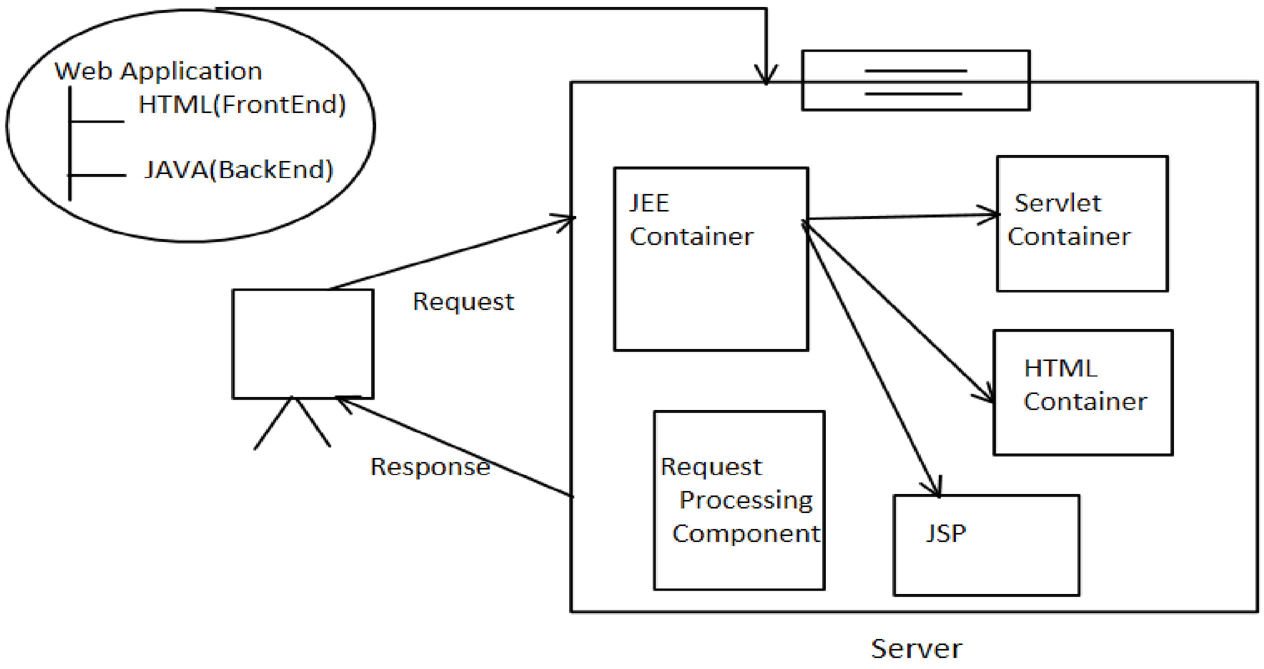

server
1)request
2)response is generated at runtime
Client
3)response is sent to the client

4. Programmer is giving an application to the server technically this process is referred as Deployment.
5. Deployment process is carry out once after development of application.
6. There are two types of Deployment
   a. Manual deployment.
   b. Automated Deployment.
7. In manual deployment programmers are responsible for writing the code as well as giving the application to the server.
8. In automated deployment programmers are responsible to writing the code whereas deployment is carried out by Maven Tool.

Note: Developing the web application means writing the code for both frontend & backend of an application.

JEE Container

1. JEE container is an important component within the server which manages entire web application.
2. The functionalities perform by JEE container are-
   a. Once after deployment JEE container will segregate the files of an application into separate container the based on the extensions of the files, which results in faster response.
   b. Once after segregation of files user can make an request & JEE container provide an response. This process is technically referred as processing of users request.
   c. JEE container manages servlet life cycle.
   d. JEE container responsible for the creation of config object & context object.
   e. JEE container handles Servlet chaining.



Type of Project

1. In eclipse there are multiple types of projects can be created based on requirement
   a. Java project
   b. Dynamic web project
2. Programmers prefers java project if requirement consist of only backend files.
3. Programmers prefers dynamic web project if requirement consist of both frontend & backend file.
4. Dynamic web project consist of four important folders.
   a. Src folder
   b. Web content folder
   c. Build folder
   d. Lib folder
5. Build folder is basically used for testing an application.
6. Source folder is use for backend file whereas web content folder is use for frontend file.
7. Lib folder contents jar files which is required to run the program.

Types of Classes

1. We have many types of classes which can be created in the project.
   a. Normal Java class.
   b. Servlet class
2. A class is said to be a normal java class if the execution of class starts with main method.
3. A class is said to be a servlet class if execution of class starts with request from HTML file.

Date Class

1. It is used to print current date & time in java standards & present in java.util package.
2. SimpleDateFormat class is used to print current Date & time in the proper order which is specified in constructor.
   Where,
   h ---> hours
   m ---> minutes
   s ---> seconds
   d ----> current date
   M ----> current month
   y ---> current year
3. Format() is used to convert current date & time into proper specified format.

Generic Servlet

1. It is an abstract class present in javax.servlet package, which contains one abstract method called it as service().
2. This method takes two arguments
   a. ServletRequest ----> it provides user request information to server.
   b. ServletResponse ----> it represent response send back to user.
3. Syntax.

```
class Sample extends GenericServlet
{
        @Override
        + void service(ServletRequest req, ServletResponse resp)
        {
                //Backend code
        }
}
```

4. All the servlet classes will be executed once after there is an request from HTML file.
5. The basic way of making request from html file is by creating an button.
   <input type = "submit" value = "checkDate">

To get servlet-api jar file :

c: ----> program files ----> apache tomcat foundation ----> tomcat 8.0 ----> lib ----> servlet-api.jar

Que. How to Link front-end file and back-end file in dynamic web project?

Ans- We can link front-end file and back-end file by providing two important information.

1. In html file we have to provide form action
   <form action= "Servlet Class Name"></form>
2. Above the servlet class, we have to provide @webservlet annotation with the slash.
   @webservlet("/servletClassName") ----> relative path.

Fetching user information from HTML file.

- To fetch user information from HTML file to servlet class programmer have to provide two important steps.

  Step1 :
  Declare identifier for each input in html file by using 'name' attribute.

  ```
  <input type = "email" name = "email"></input>
  ```

  Step2:
  - In the servlet class programmer have to call getParameter().
  - As an argument for getParameter(), we have to pass identifiers which are declare in HTML file.
  - The return type of this methods is String type.
  - It called by using ServletRequest reference variable.

  ```
  String email = req.getParameter("email");
  ```

Note: For giving the user information from html page to Database through servlet class we have to use static method forName() which is present in class "Class" . It is used to register the driver with servlet class.

Syntax - Class.forName("com.mysql.jdbc.Driver");

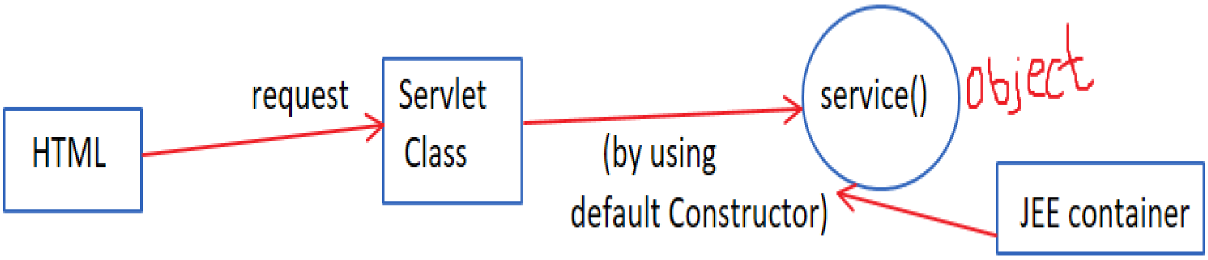
PrintWriter

- Printing the output on to the browser, programmers using PrintWriter Class which is present in java.io package.
- The helper method for print writer class is getWriter(). To print the contents on to browser, we make use of println().
- Syntax : PrintWriter writer = resp.getWriter();

Servlet Life Cycle

1. Servlet life cycle shows all the phases from the time of request and till servlet class provide the response.
2. The main purpose of servlet life cycle is pass the values which varies periodically to the application.
3. In servlet life four phases are there:
   a. Instantiation phase
   b. Initialization phase
   c. Service phase
   d. Destruction phase

4. Instantiation phase
   a. In this phase servlet object is created by the container.
   b. When user makes a request to the servlet class, container converts servlet class into servlet object by using default constructor present in servlet class.
   c. If servlet class does not contain default constructor then we get InstantiationException (unchecked exception)



5. Initialization phase
   a. The process of passing the value for varies is referred as Initialization of servlet class.
   b. In this phase, programmer will be passing the values which are varies.
   c. Passing the values can be done in two different ways.
      i. By using **init parameter**
      ii. By using **context parameter**
   d. The values which are passed for one servlet class is referred as init parameter. It is having **local scope.**
   e. The values which are passed for all the servlet classes in the entire application referred as context parameters. It is having **global scope**.

   Config & Context Object

   6. Config object & context object are created by container during initialization phase.
   7. As a programmer we use config object to pass init parameter and context object to pass context parameter.
   8. Since, config object make changes for one servlet class, the scope of config objet is local scope.
   9. Context object make changes for all servlet classes of entire application the scope of context object is global scope.
   10. For web application, container creates one context object and n no. of config objects where n is equals to no. of servlet classes.

6. Service Phase

   a. In this phase request and response object is created by service method with the help of container.

b. When first user makes request , container converts servlet class into servlet object using default constructor present in servlet class.

c. If there are 100 user request for 1 servlet class then 1 servlet class object and 100 request & response object will get created.

100 servlet class present in application & 100 separate user send the request for each servlet class.
Servlet object no ----> 100
Request ---> 100(user) * 100(servlet) = 10000 request object
Response ---> 100 * 100

d. If there are 100 user request for 1 servlet class then only 1 servlet class object & 1 request & response object will get created then this kind of application we are calling it is single threaded application.

Note:

a. by default servlets will act as an multithreaded as a programmer we can make servlets to act as single threaded by using synchronized keyword.
b. If container working with multithreaded applications then each user will be maintain separately by creating multiple request object & response object. This results in independent behaviour.
c. If container is working with single threaded application then container creates one request object and one response object which can be use by only single user at time this results in dependent behaviour.

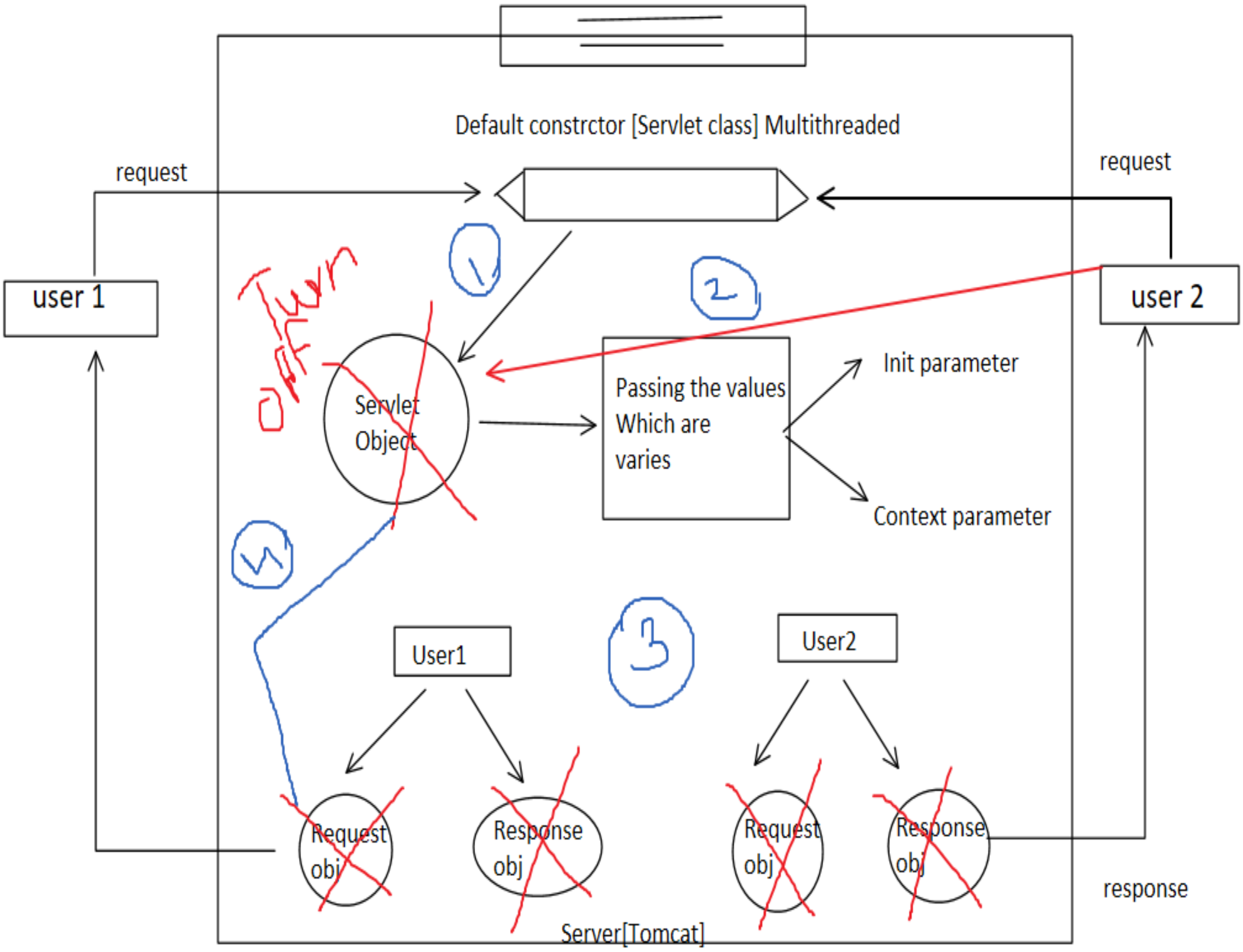
Conclusion ---->

a. No. of servlet class = no. of servlet object ---> 1st phase
b. No. of servlet class = no. of Config object & only one Context object for application. ----> 2nd phase
c. No. of user request = no. of request & response object for multithreaded application
   No. of servlet class = no. of request & response object for single threaded application ----> 3rd phase

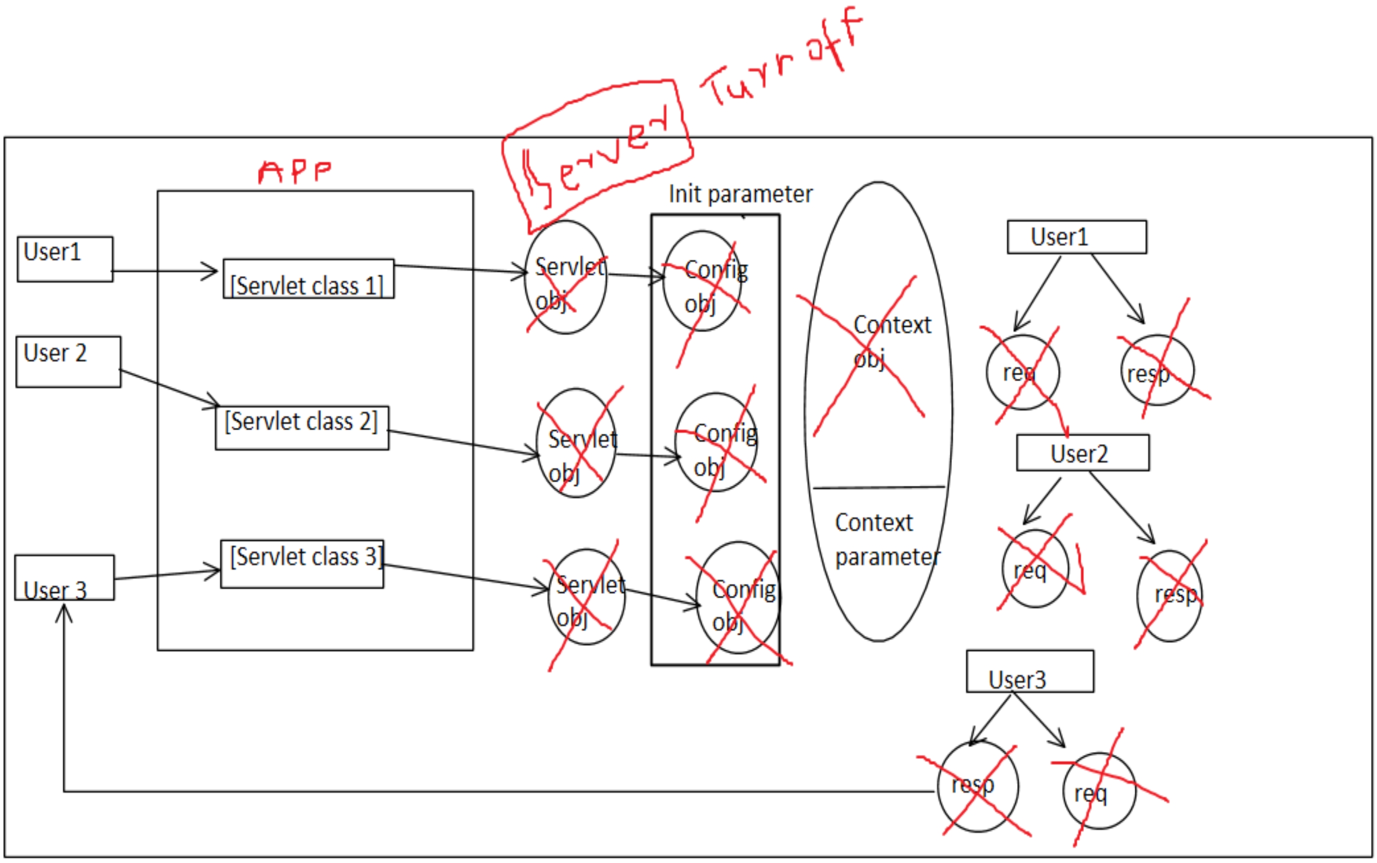
g. Destruction phase

a. In this phase the objects which have created in the above phases will be destroyed by the container.
b. Once after objects are destructed servlet life cycle ends.

Note: servlet object will get destroy when server gets turn off.



Default constrctor [Servlet class] Multithreaded
request
request
user 1
user 2
Turn off
1
2
3
4
Servlet Object
Passing the values Which are varies
Init parameter
Context parameter
User1
User2
Request obj
Response obj
Request obj
Response obj
response
Server[Tomcat]



Server Turn off
APP
Init parameter
User1
User 2
User 3
[Servlet class 1]
[Servlet class 2]
[Servlet class 3]
Servlet obj
Servlet obj
Servlet obj
Config obj
Config obj
Config obj
Context obj
Context parameter
User1
req
resp
User2
req
resp
User3
resp
req

Working with Config object:

1. To pass the init parameter in class programmers using Config object.
2. To work with config object follows the below steps:
    a. Use @WebServlet above class. In this we have make use of two attributes mentioned as below.
        i. **urlPatterns** ---> to pass relative path of servlet class.
        ii. **initParams** ----> to specify init parameter for servlet class.
        iii. **@WebInitParam** annotation is used to create init parameter.

    b. Syntax :

    ```
    @WebServlet(urlPatterns = "/servletClassName", initParams =
    {@WebInitParam(name = "id" , value = "value")})
    ```

    1. Now, override **+ init (ServletConfig config)** method in servlet class to use init parameter in program. This method is used to get the config object in form of argument.
    2. In above method call **getInitParameter()** which present in **ServletConfig** interface by using its reference variable. This method is used to get value of init parameter from config object. The return type of method is **String**.
    3. Syntax :

    ```
    + init (ServletConfig config)
    {
            String value = config.getInitParameter("id")
    }
    ```

HTTP SERvlet

1. There are two types of method request in HTML File
    a. Get
    b. Post
2. If method = "post" then user information is hidden without displaying on url. Hence post request is secure.
3. If method = "get" then user information is displaying on url. Hence get request is unsecure.
   Note: by default, if "method " attribute is not declared in html then it works as get request.
4. HTTP servlet is an abstract class present in javax.servlet package.
5. Being abstract class it does not contains any abstract method instead it contains complete methods. Those complete methods are doGet(HttpServletRequest req, HttpServletResponse resp) & doPost(HttpServletRequest req, HttpServletResponse resp)
6. If method = "post" in html file then override doPost() in HttpServlet class.

7. If method = "get" in html file then override doGet() in HttpServlet class.

   Note:
   1. If there is mismatch between method in HTML file & method overridden in HttpServlet then its results in 405 error.
   1. doPost() & doGet() are declared as protected which makes inheritance mandatory.
   2. If class contains super keyword then its results in 500 error/405 error.

Servlet Chaining

1. As a programmer we use servlet chaining to link one functionality to another functionality sequential flow of execution.
2. When a user make a request, servlet class can execute the logic either by using service methods, doPost() or doGet() and then communicate to other resources like HTML files, JSP files and another servlet class.
3. This type of servlet communication to other resource is referred as servlet chaining.
4. Servlet chaining can be performed in two ways.
   a. By using RequestDispatcher Interface
   b. By using sendRedirect()
5. RequestDispatcher Interface
   a. It is present in javax.servlet package
   b. It is used to chain from one servlet class to other resources like HTML file, Jsp file or another servlet class.
   c. Syntax -

      RequestDispatcher dipatcher =
      request.getRequestDispatcher("filename/relative path");

   4. In the RequestDispatcher one important method is present -
      Include(req, resp) performs servlet chaining in forward direction if it is valid & backward direction if it is not valid.
6. sendRedirect()
   a. It is present in HttpServletResponse interface.
   b. It is used to chain one servlet class to other resources which is present in same project or in external project.
   c. As an argument programmer can pass relative path / absolute path/ url.
   d. Syntax:
      resp.sendRedirect("relative path / absolute path")

Session

1. It is time interval for user between login & logout.
2. Sessions are used to store user information in temporary format.
   Note: Database is store user information permanently.

3. Whenever application needs user information to perform any task, so that time programmer can access it from session instead of DB and this results in faster execution.
4. When user logged in into application then information is stored in session object from database and when user logged out from application then information is deleted from session.
5. To store session object programmer using HttpSession interface rv & to create object we have to use getSession() which present in HttpServletRequest.

   Syntax. - HttpSession session = req.getSession();

6. To store user information into session object programmer using setAttribute() present in HttpSession interface. This method takes two argument 1) identifier for user information.
   1. User information which we want to store.

   For example - session.setAttribute("stuName", name);

7. By using Session programmers can reduced the execution and save the memory.
8. To give the time interval for session used setMaxInactiveInterval(int sec) which accepts arguments in seconds.
9. To retrieve values from session object in program used following method.

   ```
   session.getAttribute("identifier of user info");
   ```

   When programmers wants to retrieve value from session that values are already upcasted into the object class. Hence firstly downcast the value and then use it.

   For example ---> String name = (String)session.getAttribute("identifier");

10. After the session time interval all the object which stored in session that gets null. So that time programmers can close the session by using invalidate() which present in HttpSession interface.

    For example ---> session.invalidate()

Java Server Pages

1. The main idea of JSP files is to avoid the disadvantage of Printwriter class.
2. While developing applications, programmers will be having a requirement of giving the output which is the combination of Java code and html code.
3. But PrintWriter class is having the ability of giving the output of java coding only.
4. To overcome the above drawback the programmer will make use of JSP files.
5. To differentiate java coding & html coding in the jsp file, programmer using jsp elements.
6. There are five types of JSP elements.

   a. Expression
   b. Declaration
   c. Action
   d. Directive
   e. Scriplets.

7. Scriplets :

   a. Programmers will make use of this element to provide local contents in the jsp file.

   b. It is indicated by <% -JC- %>

h. Expression:

   a. Programmers will make use of this element to print java contents on the browser.

   b. It is indicated by <%= -JC- %>

Note: All expression jsp elements contents will not end with semicolon(;).

JSP Life Cycle:

a. In Jsp life cycle we have five phases.

   a. Translation
   b. Instantiation
   c. Initialization Servlet life cycle
   d. Service
   e. Destruction

b. Translation phase

   a. In this phase all the jsp files will be converted into servlet class, converted servlet classes is technically referred as "Translated Servlets"(T-servlets).

   b. All the jsp files are converted into servlet class at runtime that means only after execution of JSP file.

   c. Translated servlet file names will similar to jsp file name.

   Example - If jsp file name is Demo.jsp then Translated servlet would be Demo_jsp.java.

d. During runtime all JSP files are firstly converted into .java file and later compiled for .class file of T-servlets.
e. After compilation of T-servlets, the servlet lifecycle process starts.

Note. Instantiation, Initilization, service and destruction phase of JSP life cycle is similar to servlet life cycle.

Path of Translated servlet :
D:\WorkSpaceFolder\.metadata\.plugins\org.eclipse.wst.server.core\tmp0\work\Catalina\localhost\ProjectName\org\apache\package-name

3. Declaration:
   a. Programmers will make use of this element to provide global contents within JSP file.
   b. It is indicated by <%!   -JC- %>

4. Action JSP element

   a. As a programmer action jsp elements allows programmers to perform Java task with the appearance of HTML tags.
   b. Syntax:
      <jsp:action-name attribute = "value"></jsp:action-name>
   c. There are many JSP action tags, some of them are

      i. useBean
      - It is basically use by the programmers in order to create an object of a class.
      - It takes three arguments.
         1. Id :- represents reference variable of an object.
         2. Class :- represents the restriction on the object it takes four values.
      - If scope = "page" then the object created is accessible only in the current JSP files.
      - If scope = "application" then the object created is accessible only in the entire application.
      - If scope = "request" then the object created is accessible only in the current JSP files as well as immediate JSP file.
      - If scope = "session" then the object created is accessible only till particular time interval.

Note: If a scope is not represented for <jsp:useBean> tag then by default scope is considered as "page".

For example : <jsp:useBean id = "c1" class = "org.jsp.app.car" scope = "page"></jsp:useBean>

      i. setProperty

- It is basically use for initializing the object using action tag. It takes three important attribute.
  1. Property :- It represents variable name of an object which has to be initialize.
  2. Name :- It represents reference variable name of an object.
  3. Value : represents actual information which has to be passed for the variables.

ii. getProperty

- It is basically use to retrieve the information from the object. It takes two attribute.
  1. Property
  2. Name
- Forward
- It is basically used to link one jsp file to another jsp file. It takes one important attribute.
  1. Page :- represents jsp file name to which chaining has to be perform.
- Include
- It is used to include one jsp page to another one for achieving code reusability and to link one jsp file with another. It takes one important attribute.
  1. Page :- represent jsp file name which code has to be include.

Note: for include action tag, we have to pass only the file names which are existing. If in case we pass file name which is not existing then it throws compilation error.

for forward action tag, if we pass file name which are not existing then it throws runtime exception.

- Directive JSP element
  1. JSP directives basically takes the help of other packages, other files to overcome from errors appearing in JSP file.
  2. It is represented as <%@ -JC- %>
  3. We have three types of directives
     i. Page
     ii. Include
     iii. Taglib
  4. Page Directive
     i. It mainly used to make changes in current jsp file.
     ii. Represented as given below -->

        <%@ page %>

     - Few attribute which can be used along with page directive are --->
       1. Import - used to import the files from differentiate package into current jsp file.
       2. Extends - used to extends super class members in current jsp file. We have to pass fully qualified name for this attribute. Once After extending

the class programmer can override methods of super class by using declaration.

Note: Programmer can extend only user-defined servlet class.

- errorPage -
  1. It is used to maintain JSP file name which consists of recovery code.
  2. If given jsp file consist of an exception then control will be automatically shifted to the jsp file name which is consist of recovery code.
- isErrorPage -
  1. It is used to mention the current JSP file consist of only recovery code.
  2. It is always writes to indicate the jsp file which only consist of recovery code.

- Include Directive
  1. It is basically mentioned within body tag.

```
<body>
	<%@ include file = "filename.jsp" %>
</body>
```

- This helps the programmer use multiple file together, where all the file output will be displayed at once on the browser.

f. Taglib directive

i. It is used to create custom jsp elements which can be used only if given jsp elements is not working in the application.

ii. Taglib is advance one for that we need to go through sprinMVC.(Model View Controller)

iii. We use TLD(Tag library Discriptor) file to define the tags.

Cookies

1. Cookie is a piece of information which is used to store user information within the browser.
2. There are two types of Cookies-
   a. Persistent
   b. Non-Persistent
3. The cookies which must be removed explicitly is referred as Persistent Cookie whereas The cookie which is active particular time spam is referred as Non-Persistent Cookie.

Note: Cookies are created within server and it is stored within the browser.

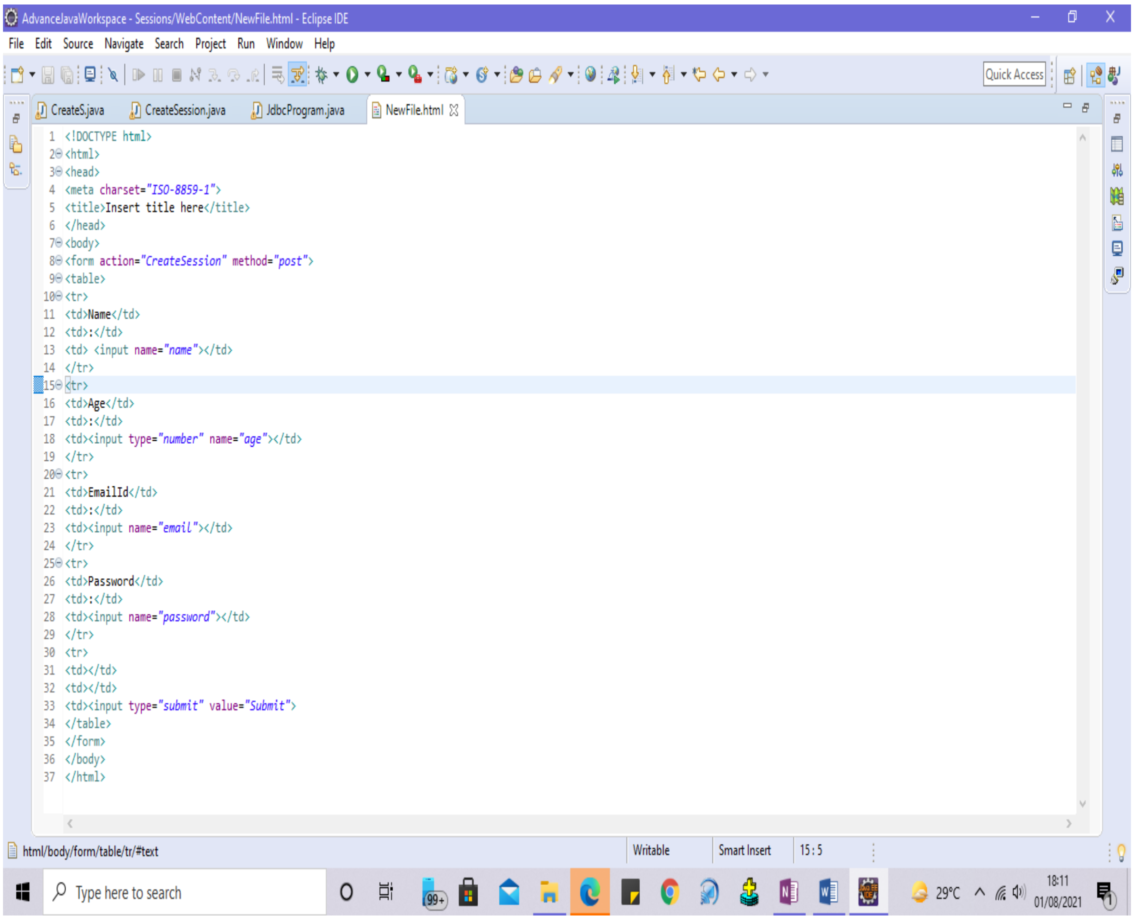
4. As a programmer if we have to create cookies then we have to make use of Cookie class, which is present in javax.servlet.http package.
5. The syntax of creating cookies is –

Cookie refvar = new Cookie(String identifier, String userInfo);

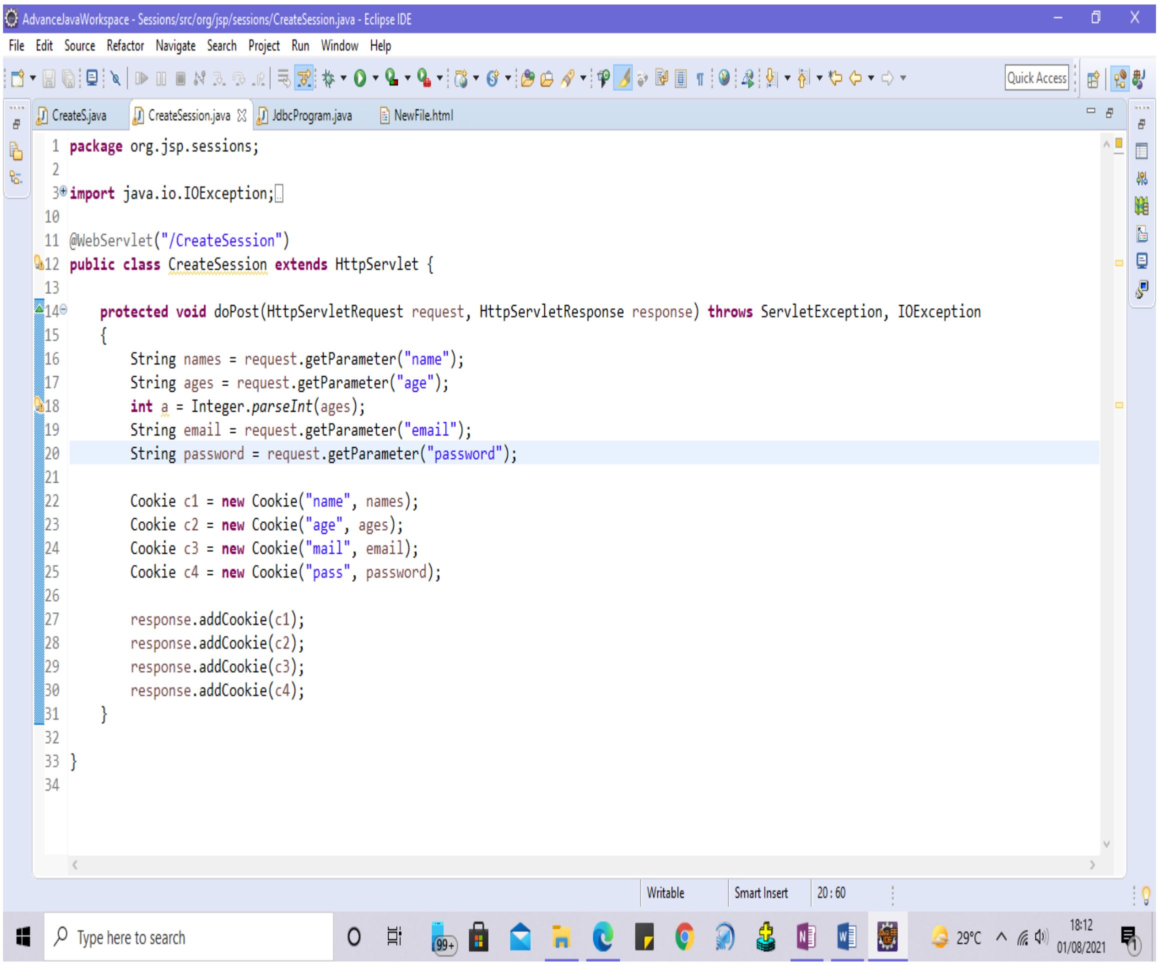
Where, identifier reference to the Cookie which is stored within the browser.

6. To send that cookies from server to browser, we have to call addCookie(), which must be called by using respose ref var.

Example: resp.addCookie(cookie1);


AdvanceJavaWorkspace - Sessions/WebContent/NewFile.html - Eclipse IDE
File Edit Source Navigate Search Project Run Window Help
Quick Access
CreateS.java
CreateSession.java
JdbcProgram.java
NewFile.html
html/body/form/table/tr/#text
Writable
Smart Insert
15 : 5
Type here to search
29°C
18:11
01/08/2021

```html
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<form action="CreateSession" method="post">
<table>
<tr>
<td>Name</td>
<td>:</td>
<td> <input name="name"></td>
</tr>
<tr>
<td>Age</td>
<td>:</td>
<td><input type="number" name="age"></td>
</tr>
<tr>
<td>EmailId</td>
<td>:</td>
<td><input name="email"></td>
</tr>
<tr>
<td>Password</td>
<td>:</td>
<td><input name="password"></td>
</tr>
<tr>
<td></td>
<td></td>
<td><input type="submit" value="Submit">
</table>
</form>
</body>
</html>
```



AdvanceJavaWorkspace - Sessions/src/org/jsp/sessions/CreateSession.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
CreateS.java
CreateSession.java
JdbcProgram.java
NewFile.html
Writable
Smart Insert
20 : 60
Type here to search
29°C
18:12
01/08/2021

```java
package org.jsp.sessions;

import java.io.IOException;

@WebServlet("/CreateSession")
public class CreateSession extends HttpServlet {

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException
    {
        String names = request.getParameter("name");
        String ages = request.getParameter("age");
        int a = Integer.parseInt(ages);
        String email = request.getParameter("email");
        String password = request.getParameter("password");

        Cookie c1 = new Cookie("name", names);
        Cookie c2 = new Cookie("age", ages);
        Cookie c3 = new Cookie("mail", email);
        Cookie c4 = new Cookie("pass", password);

        response.addCookie(c1);
        response.addCookie(c2);
        response.addCookie(c3);
        response.addCookie(c4);
    }

}
```

Note: We can store userinformation inside String array and then pass that rv to Cookie.

7. getCookies() is used to retrieve cookies from browser which must be called by using request ref var.
8. The return of getCookies() is Cookies[], which represents multiple cookies.
9. To retrieve the values from Cookie we have to make use getValue().

Note: We cannot use one jsp element into another jsp element.

10. As a programmer we can decide the time span of an cookie by using setmaxAge() which accepts integer values in seconds and called by using cookies ref var.

For example:

```
Cookie [] carr = request.getCookies();
For(Cookie cks : carr)
{
        System.out.println(cks.getValue());
}
```

Note: To view stored cookie in client machine follow below steps:

1. Go to chrome settings
2. Cookies and other site data
3. See all cookies
4. Search cookies -→ localhost.