Date:__.__.____

Advance Java.

JDBC : Java data base connectivity

JDBC Execution Flow

1. Loading driver class

used to load class manually

```
Class.forName("drivers.class.Name");
```

Built in class Class

static method of class Class.

2. Creating JDBC connection

```
DriverManager.getConnection("url", "username", "pass");
```

3. Creating / writing query statements.
   using 3 ways
   i) statement
   ii) Prepared statement
   iii) Callable statement

4. Executing Query
   i) execute();
   ii) executeUpdate(); DML
   iii) executeQuery(); DQL

5. Close the connection

```
Connection.close();
```

JDBC is a tier Non Web Application.

~~Http codes~~ :-

HTML Page creation in VSS and run on browsers.
~~HTML in~~

dynamic web Application. [static HTML Pages]

development

File → new → Dynamic web project → Java 1.8 → finish

Open project → src → main → webapp → right click → new → HTML files creation.

After all HTML coding.

deployment

right click → export → war file
Browse the location
Tomcat 9.0 ⇒ Webapps → save → Finish
startup.bat ⇒ Tomcat server starts
All projects in webapps folders gets deployed.

request from browser

go to browser ⇒ Http://localhost:8080/(our web Project)/html file name

This is 2 tier web Application.

It is impossible to create numerous, thousands of HTML's beforehand statically
so, many sites generate HTML dynamically depending on users input at runtime. -Request time

Java provides servlet for generating Dynamic responses
It is a server side java component used to generate dynamic web page.

creating servlet project:

file → new → dynamic web Project → Give name
Java 1.8 (modify)
→ next → next → check web.xml → finish
base

Open project:

> deploy ----
> JAX ----
> Java resources
v src/main/java

here we need to create our servlets in packages.

> Libraries.
> Reff ----
> build
v src
v main
> java
v webapp
> META-INF
v WEB-INF
□ Lib
• web.xml

here we need to map our servlets

```
<servlet>
<servlet-name> name </servlet-name>
<servlet-class> Package qualified class name. </servlet-class>
</servlet>
<servlet-mapping>
<servlet-name> name </servlet-name>
<url-pattern> /First </url-pattern>
</servlet mapping>
```

Browser → servelet request → container → check web.xml

→ checke servlet mapping → get url mapped servlet

→ check servlet → service method.

servlet mapping in web.xml is avoided by giving annotations.

@WebServlet("/servy") ← this url can be given.

when we are connecting database to servlet, we will get class not found exception so don't forget to copy mysql connector jar file in lib folder.

JSP

JSP internally converts written code into servlets.

JSP pages goes into web content folder not in Java resources.

v src
  v main
    > java
    v webapp → right click → new → JSP file

write the code we used to write in servlets in pw.println("<html----/>") and internally it will be converted into servlets.