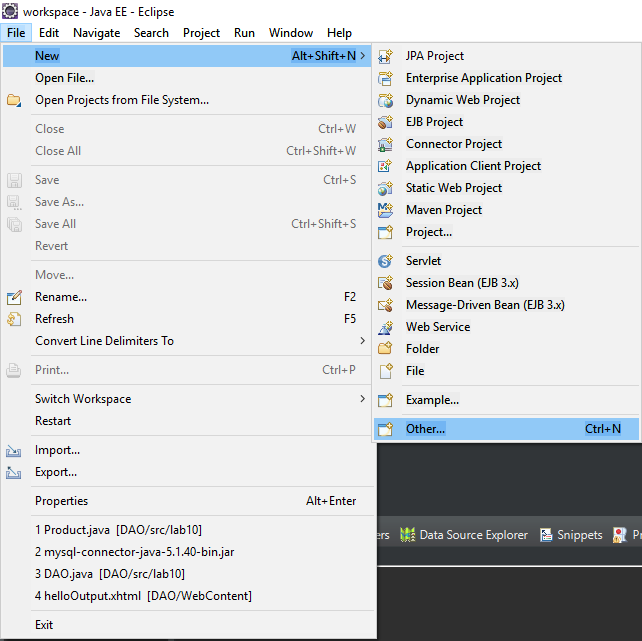
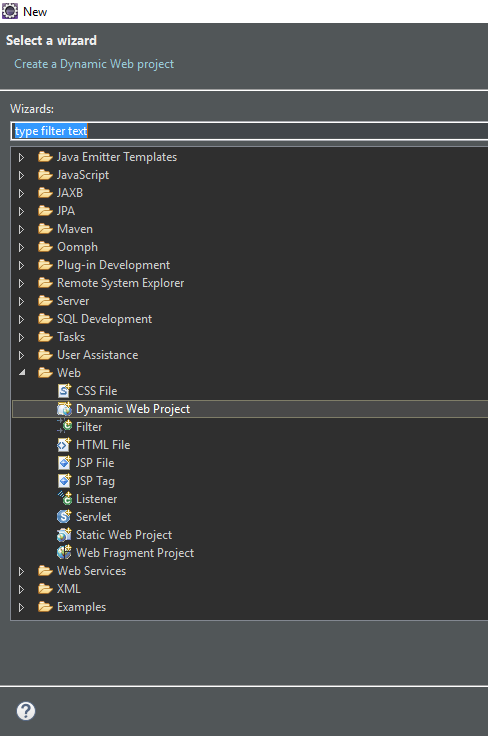
Okay so for the exam on Friday in RAD youll need Eclipse Neon.

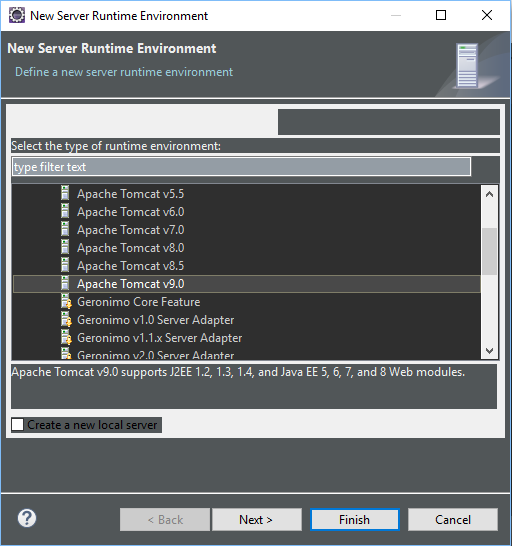
Once you open it up you’ll need to create a new Dynamic Web Project.



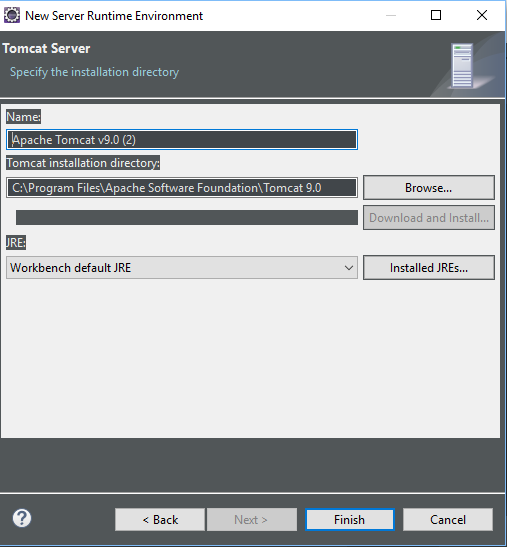
In Target runtime select new runtime if Apache Tomcat v9.0 isnt there.

Screenshot_2

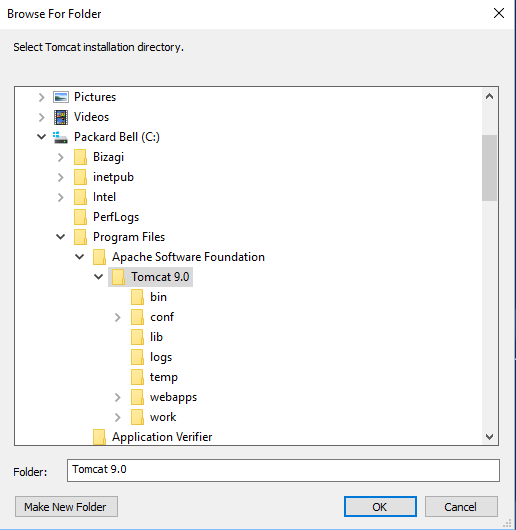
Select Apache Tomcat v9.0 and click next



Once you are creating a new runtime you select browse

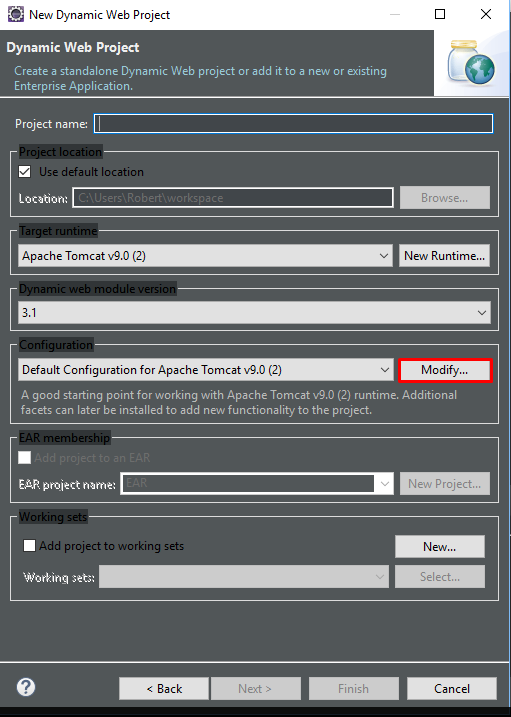


And select Tomcat 9.0 from your Apache Software Foundation

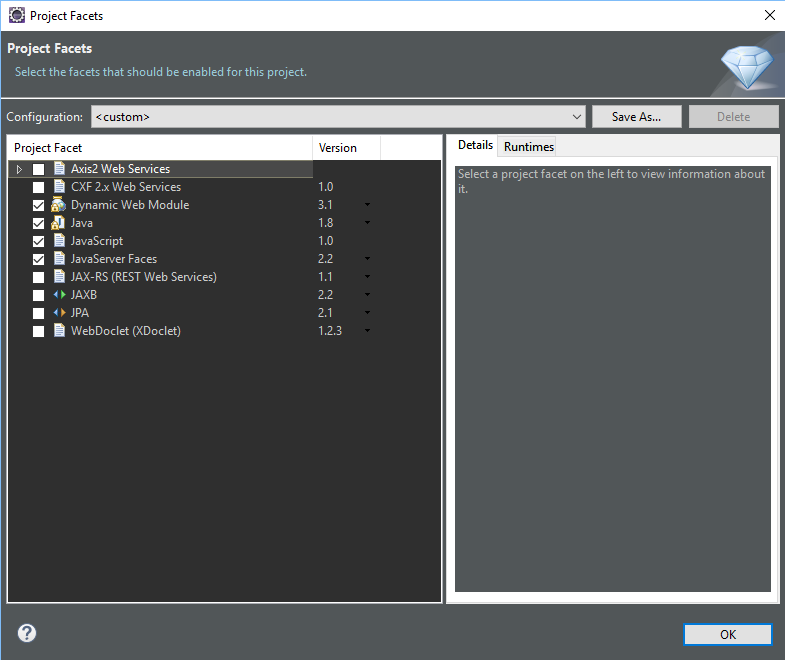


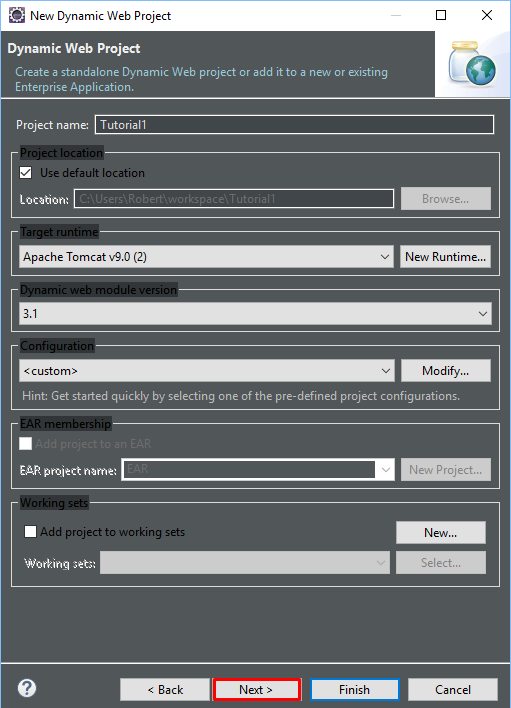
Then click next

After runtime is set to Apache Tomcat v9.0 select Modify on the configuration section

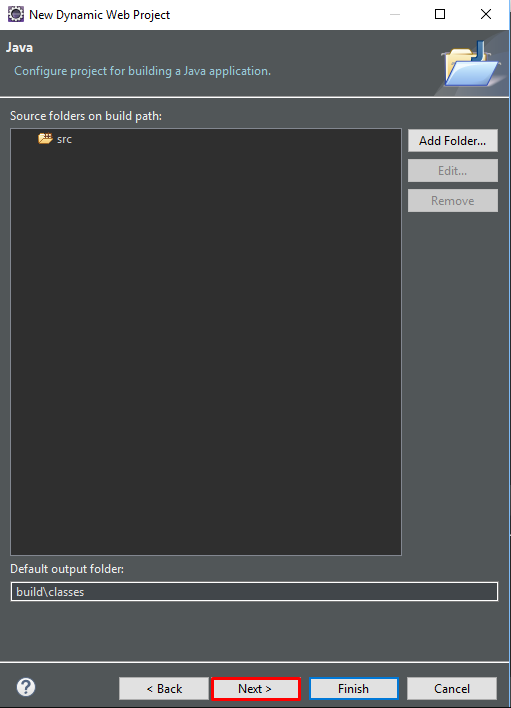


Which brings you to this window and you have to select JavaServer Faces and click okay

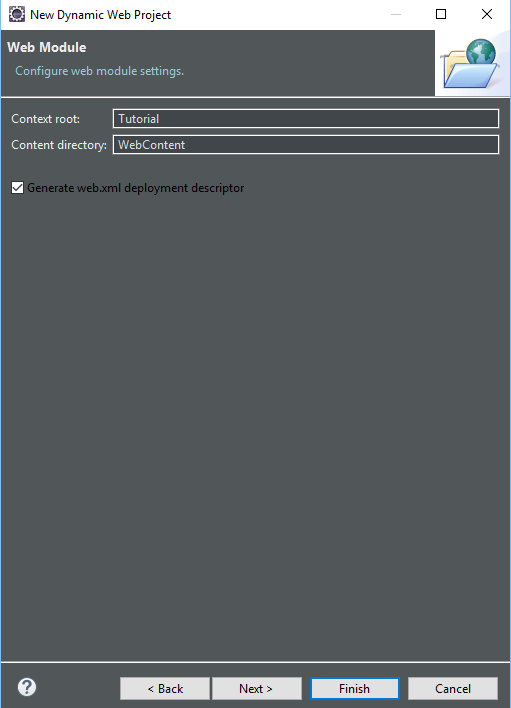


After that click next 

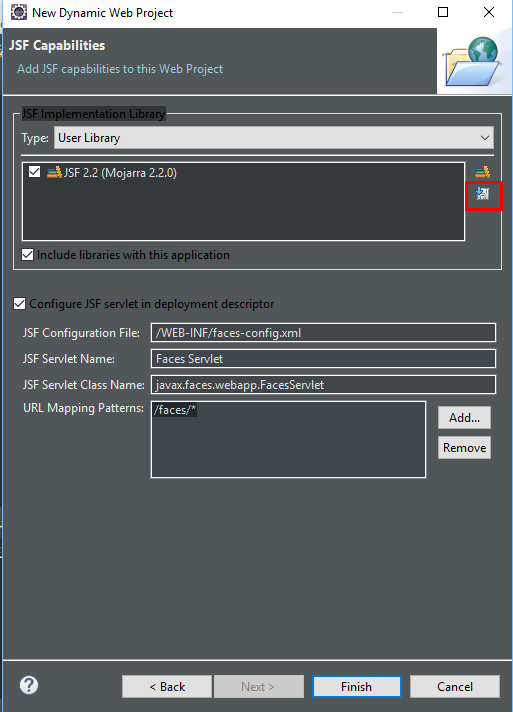
Click next again



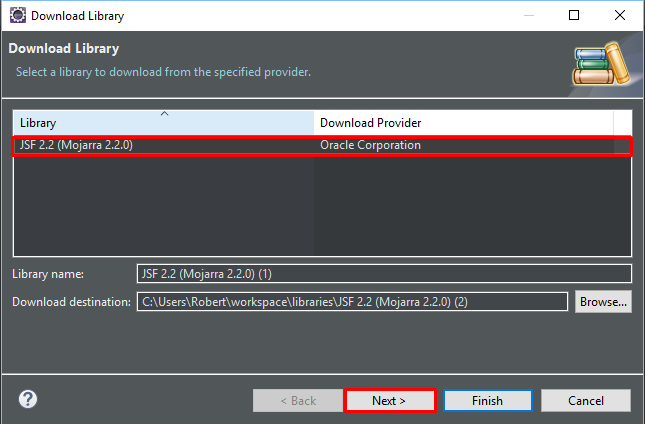
On the Web Module page select ‘’Generate web.xml deployment descriptor’’ and click next



Once you click next click the download button on the right hand side



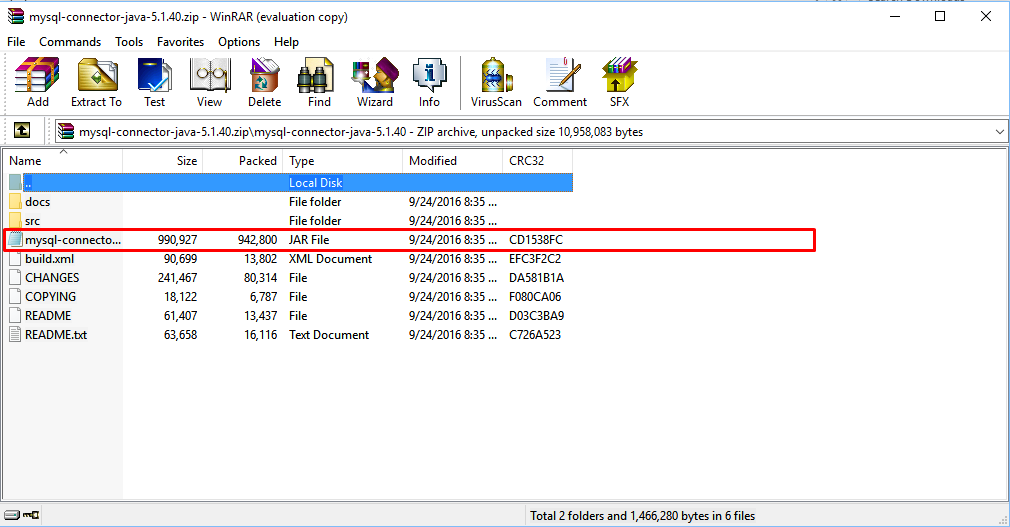
Once you click the download icon this window will open where you have to select JSF 2.2 and click next



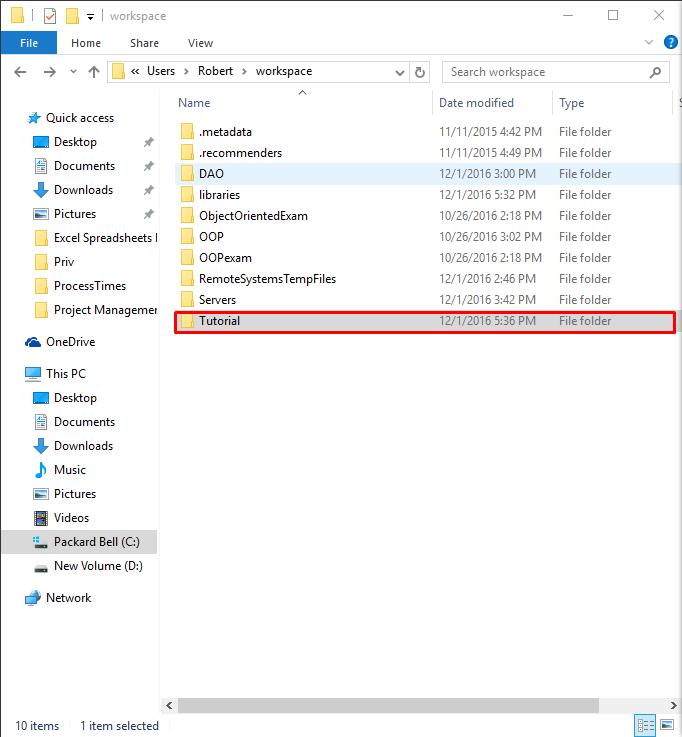
When you click next it will ask you to accept the terms and conditions so accept it and click finish , this makes the project.

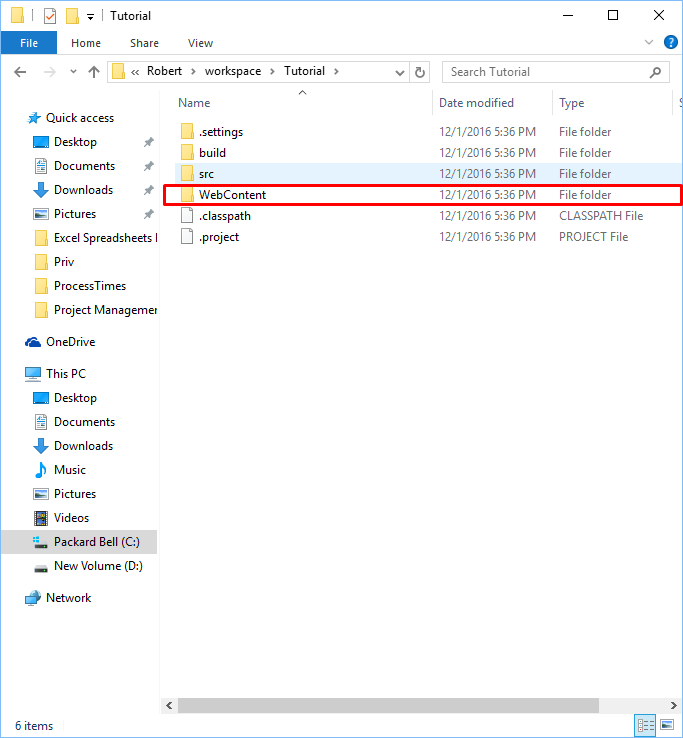
Next you’ll have to install the JDBC driver from here <https://dev.mysql.com/downloads/connector/j/>

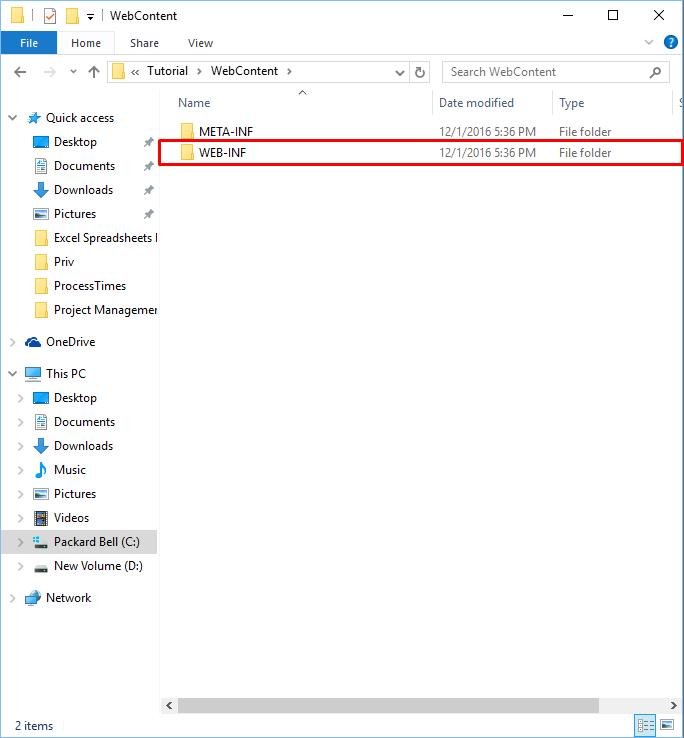
You will get this zip folder, the only file you need is the highlighted one.

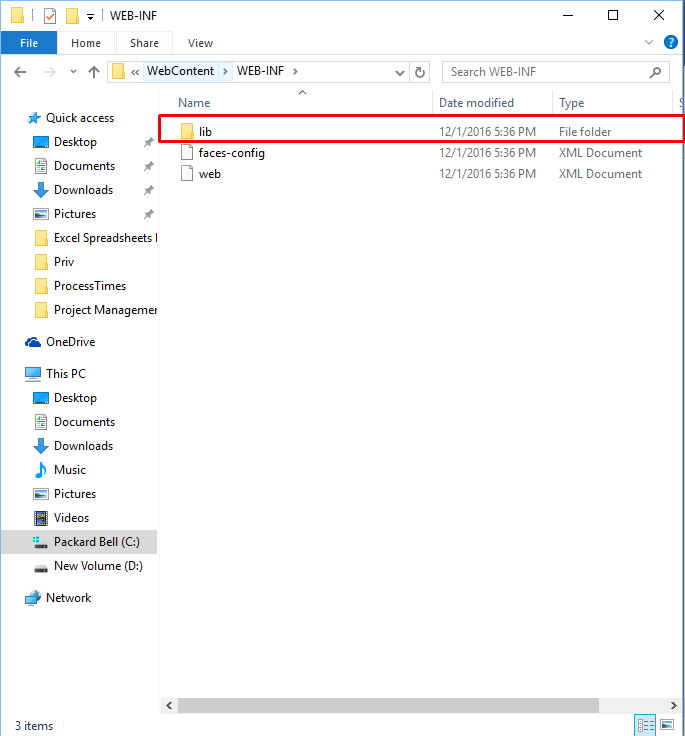


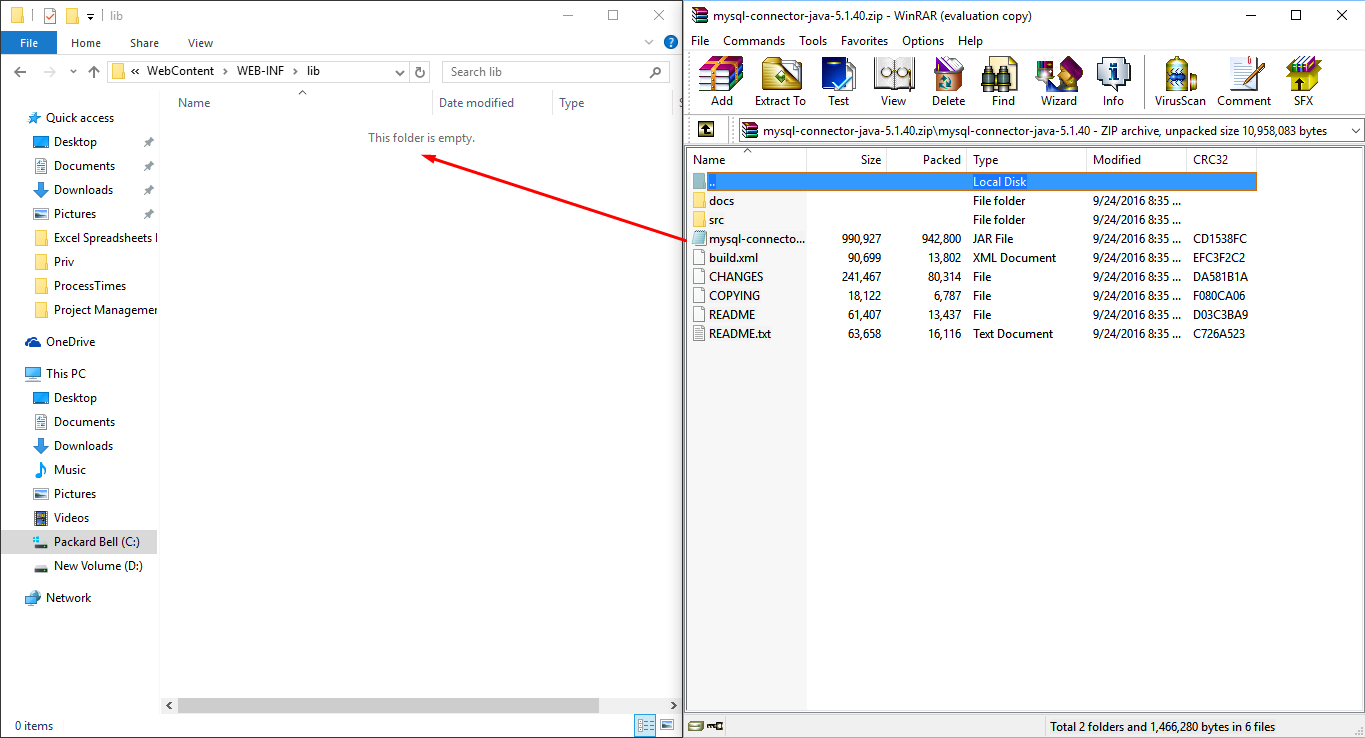
You put it into your project (wherever you save it)





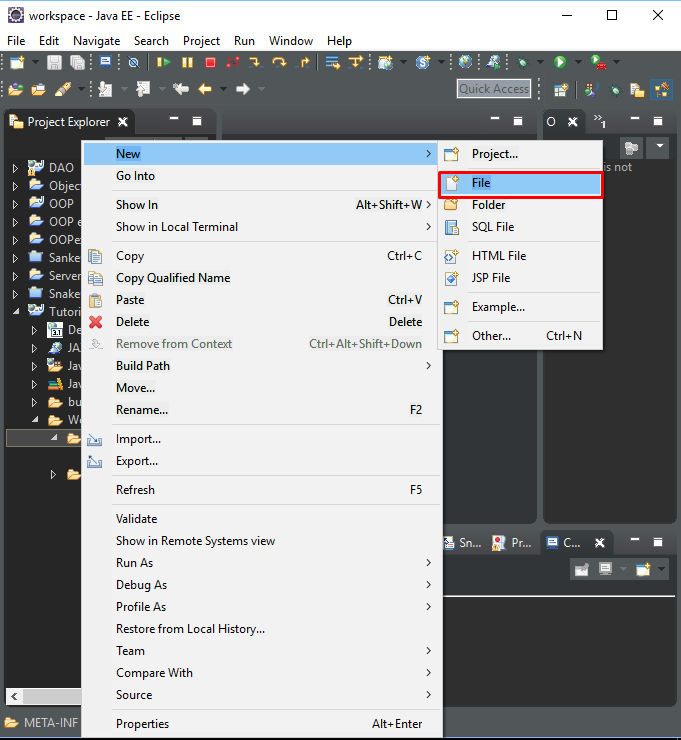




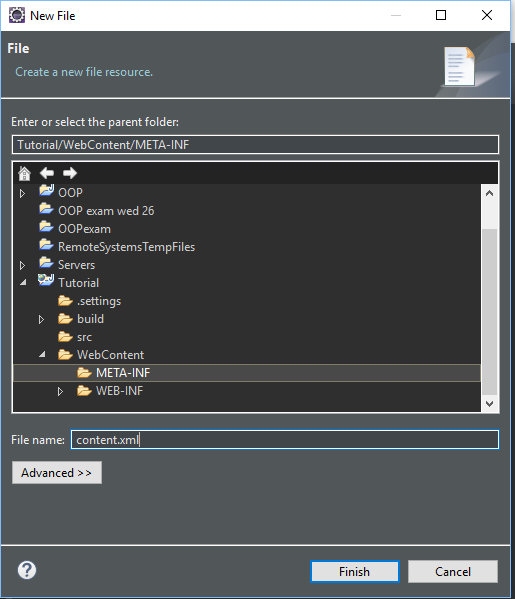


Once you’ve that setup you can start doing the code.

In the Projects Folder expand WebContent ,in this folder there are two folders META-INF and WEB-INF, right click the META-INF folder and create a new file



Call the file anything but it must be .xml



Once the file is created you open the file you just created and paste this code :

<Context>

<Resource name="jdbc/employeesdb14"

auth="Container" type="javax.sql.DataSource"

maxActive="20" maxIdle="5" maxWait="10000"

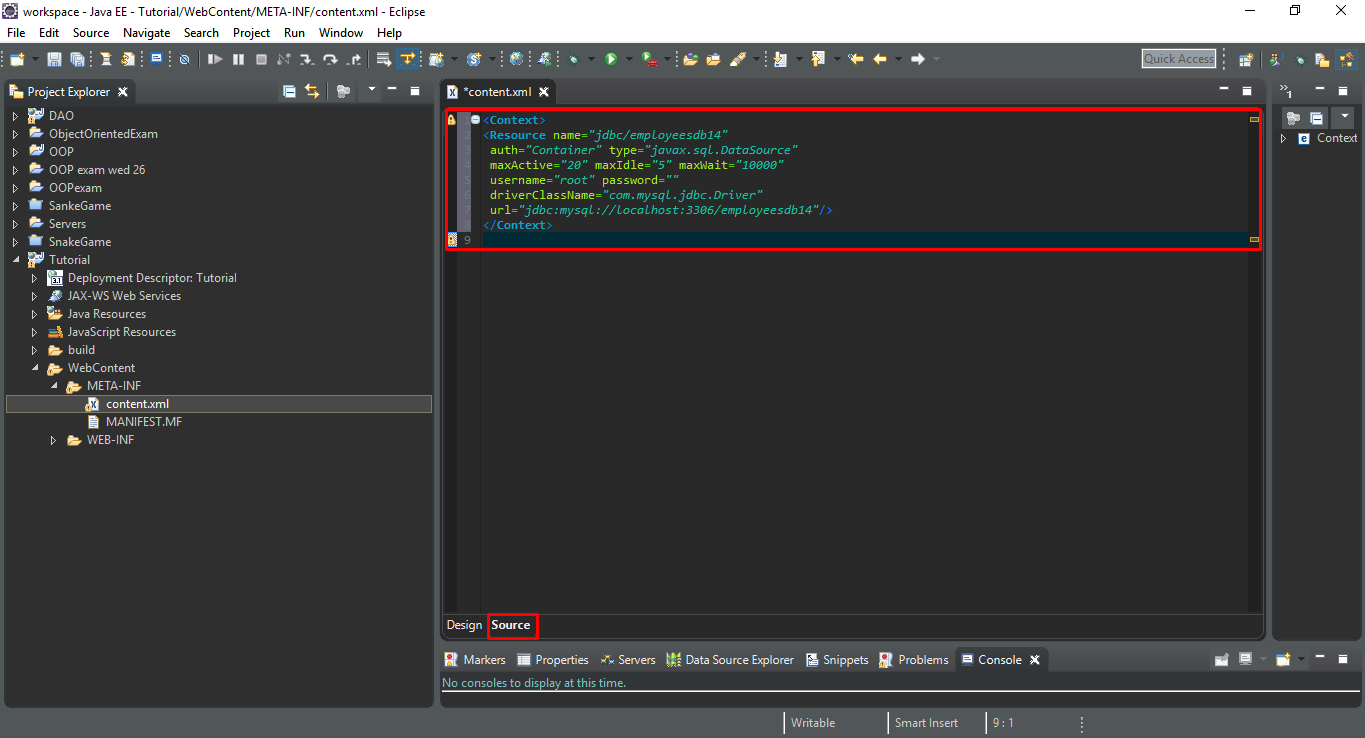
username="root" password=""

driverClassName="com.mysql.jdbc.Driver"

url="jdbc:mysql://localhost:3306/employeesdb14"/>

</Context>

Into the source tab.



After that you will need to go into the WEB-INF folder and open up web.xml file and paste this block of code into it

<resource-ref>

<description>EmployeesDB Datasource</description>

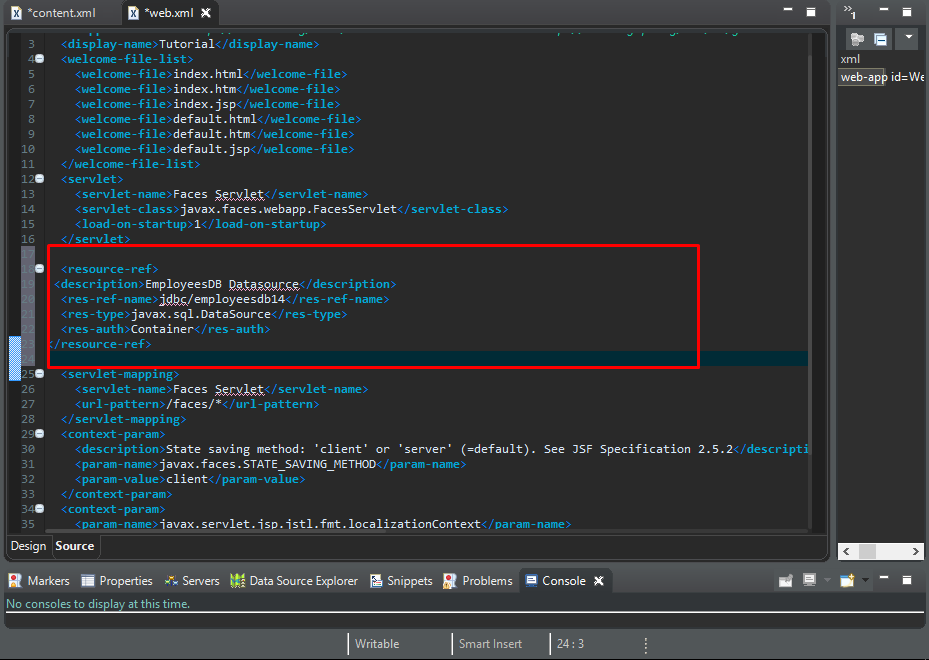
<res-ref-name>jdbc/employeesdb14</res-ref-name>

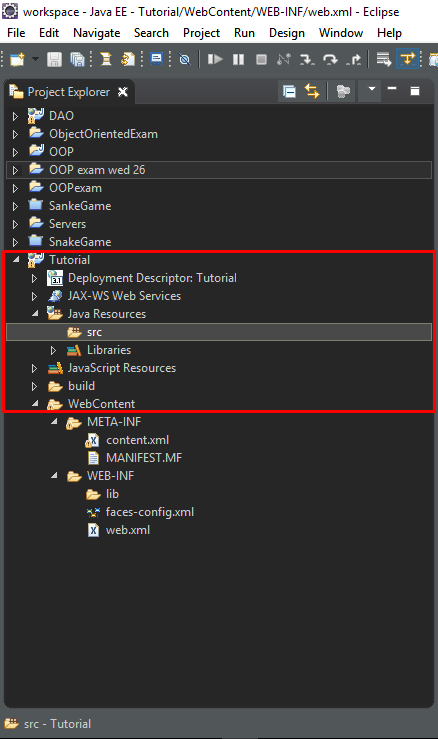
<res-type>javax.sql.DataSource</res-type>

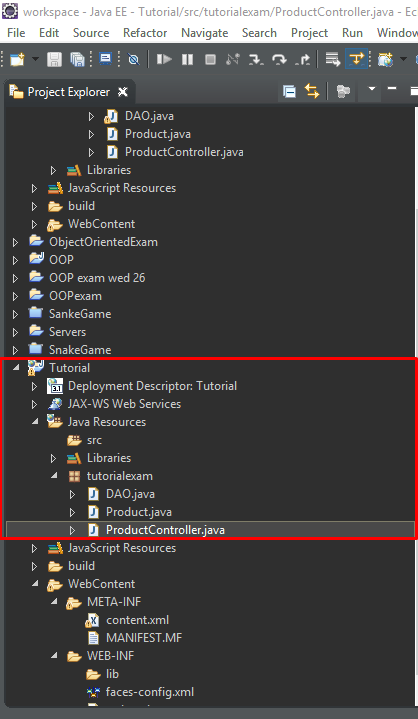
<res-auth>Container</res-auth>

</resource-ref>

This block goes in between all the other tags so don’t paste it into an existing tag



Then you create 3 java files so in the Java Resources right click source and create a package

Once the package is made make the 3 java classes call them however you want but something related to the exam.

Now for the code: in the DAO class you use this code (please edit your code and make it look more like your own as I don’t want anyone to be caught cheating so change variables etc):

**package** tutorialexam;

**import** java.sql.Connection;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.util.ArrayList;

**import** javax.faces.application.FacesMessage;

**import** javax.faces.context.FacesContext;

**import** javax.naming.\*;

**import** javax.sql.DataSource;

**public** **class** **DAO** {

**private** DataSource mysqlDS;

**public** **DAO**() **throws** **Exception** {

Context **context** = **new** InitialContext();

**String** **jndiName** = "java:comp/env/jdbc/employeesdb14";

mysqlDS = (DataSource) context.lookup(jndiName);

}

**public** **void** **deleteProduct**(**Product** p) **throws** **SQLException** {

// add pass through parameter to method for update insert & delete

Connection **conn** = mysqlDS.getConnection();

PreparedStatement **myStmt** = conn.prepareStatement("delete from product where PRODID ='" + p.getProdid() + "' and DESCRIP ='" + p.getProddesc() + "'");

// myStmt.setInt(1, p.getProdid());

// myStmt.setString(2, p.getProddesc());

myStmt.executeUpdate();

}

**public** **void** **updateProduct**(**Product** p) **throws** **SQLException** {

Connection **conn** = mysqlDS.getConnection();

PreparedStatement **myStmt** = conn.prepareStatement("update product set PRODID='" + p.getProdid() + "', DESCRIP='" + p.getProddesc() + "'");

myStmt.setInt(1, p.getProdid());

myStmt.setString(2, p.getProddesc());

myStmt.executeUpdate();

}

**public** **void** **addProduct**(**Product** p) **throws** **SQLException** {

// add pass through parameter to method for update insert & delete

Connection **conn** = mysqlDS.getConnection();

PreparedStatement **myStmt** = conn.prepareStatement("INSERT INTO product values(?, ?)");

myStmt.setInt(1, p.getProdid());

myStmt.setString(2, p.getProddesc());

myStmt.executeUpdate();

}

**public** **ArrayList**<Product> **getProductDetails**() **throws** **Exception** {

**ArrayList**<Product> **products** = **new** ArrayList<>();

Connection **conn** = mysqlDS.getConnection();

PreparedStatement **myStmt** = conn.prepareStatement("select \* from product");

ResultSet **rs** = myStmt.executeQuery();

**while** (rs.next()) {

**int** **id** = rs.getInt("PRODID");

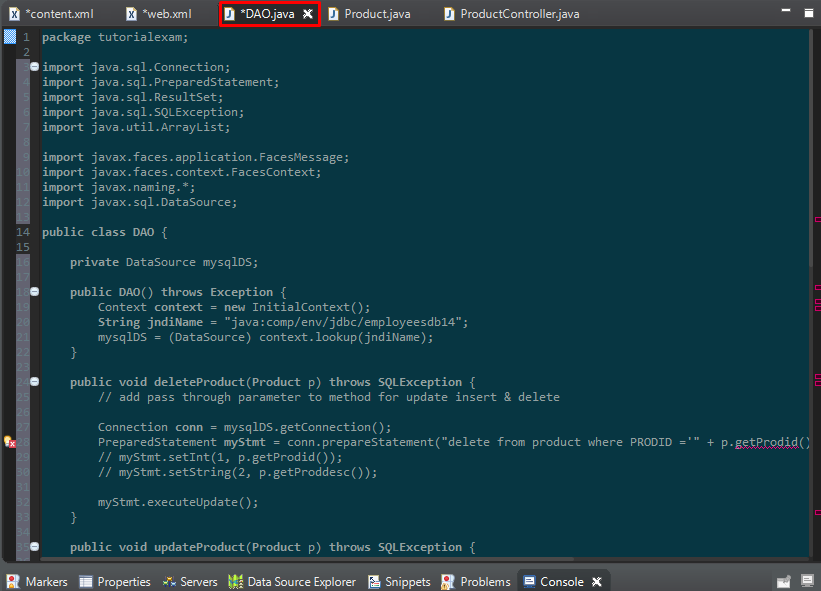
**String** **pname** = rs.getString("DESCRIP");

products.add(**new** Product(id, pname));

}

**return** products;

}



}

In product paste this(same rules as above):

**package** tutorialexam;

**import** javax.faces.bean.ManagedBean;

***@ManagedBean***

**public** **class** **Product** {

**private** **int** prodid;

**private** **String** proddesc;

**public** **Product**() {

}

**public** **Product**(**int** prodid, **String** proddesc) {

**super**();

**this**.prodid = prodid;

**this**.proddesc = proddesc;

}

**public** **int** **getProdid**() {

**return** prodid;

}

**public** **void** **setProdid**(**int** prodid) {

**this**.prodid = prodid;

}

**public** **String** **getProddesc**() {

**return** proddesc;

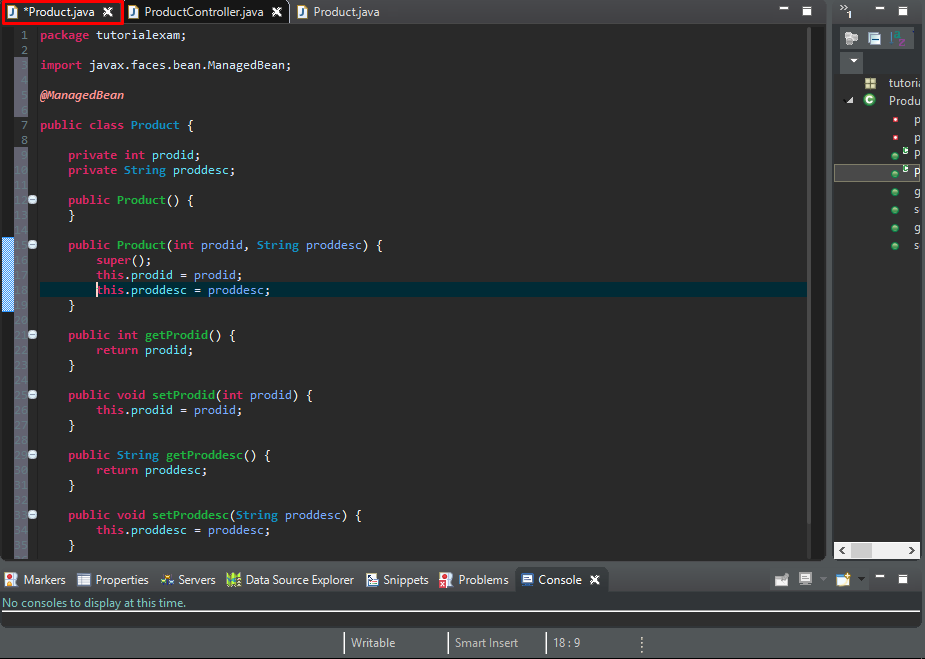
}

**public** **void** **setProddesc**(**String** proddesc) {

**this**.proddesc = proddesc;

}

}



Last part of the java is ProductController:

**package** tutorialexam;

**import** java.util.ArrayList;

**import** javax.faces.bean.ManagedBean;

//import javax.faces.event.ComponentSystemEvent;

***@ManagedBean***

**public** **class** **ProductController** {

**private** **ArrayList**<Product> products;

**private** **DAO** dao;

**public** **ProductController**() {

**try** {

dao = **new** DAO();

} **catch** (**Exception** **e**) {

e.printStackTrace();

}

}

**public** **ArrayList**<Product> **getProducts**() {

**return** products;

}

**public** **void** **loadProducts**() **throws** **Exception** {

products = dao.getProductDetails();

}

**public** **String** **addProduct**(**Product** p) **throws** **Exception** {

**try** {

dao.addProduct(p);

**return** "helloOutput";

} **catch** (**Exception** **e**) {

**return** e.toString();

}

}

**public** **String** **updateProduct**(**Product** p) **throws** **Exception** {

**try** {

dao.updateProduct(p);

**return** "helloOutput";

} **catch** (**Exception** **e**) {

**return** e.toString();

}

}

**public** **String** **deleteProduct**(**Product** p) **throws** **Exception** {

**try** {

dao.deleteProduct(p);

**return** "helloOutput";

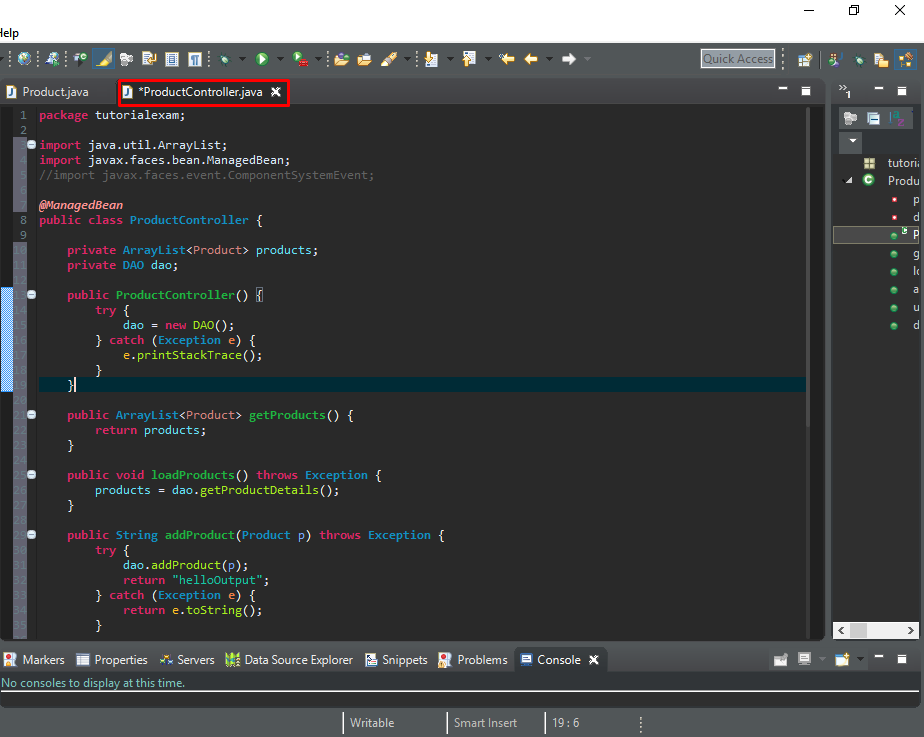
} **catch** (**Exception** **e**) {

**return** e.toString();

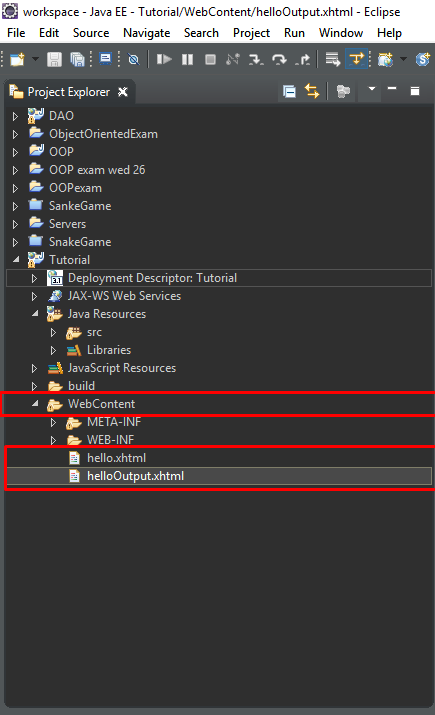
}

}

}



Java classes are finished now you need to make two more files name them anything as usual but they have to be .xhtml



In the hello .xhtml you paste this :

<!**DOCTYPE** html>

<**html** xmlns=*"http://www.w3.org/1999/xhtml"*

xmlns:h=*"http://xmlns.jcp.org/jsf/html"*

xmlns:p=*"http://xmlns.jcp.org/jsf/passthrough"*

xmlns:f=*"http://xmlns.jcp.org/jsf/core"*>

<**h:head**>

<**title**>Product Table</**title**>

<**f:metadata**>

<**f:event** listener=*"#{productController.loadProducts()}"*

type=*"preRenderView"* />

</**f:metadata**>

</**h:head**>

<**h:body**>

<**h:form**>

ProdID: <**h:inputText** value=*"#{product.prodid}"* id=*"id"*

p:placeholder=*"enter product id"* />

<**br** />

<**br** />

Prod Desc: <**h:inputText** value=*"#{product.proddesc}"*

p:placeholder=*"enter product name"* />

<**br** />

<**br** />

<**h:commandButton** value=*"load"* action=*"helloOutput"* />

<**br** />

<**br** />

<**h:commandButton** value=*"Add row"*

action=*"#{productController.addProduct(product)}"* />

<**br** />

<**br** />

<**h:commandButton** value=*"Update row"*

action=*"#{productController.updateProduct(product)}"* />

<**br** />

<**br** />

<**h:commandButton** value=*"Delete row"*

action=*"#{productController.deleteProduct(product)}"* />

</**h:form**>

</**h:body**>

</**html**>

And in helloOutput.xhtml paste this :

<!**DOCTYPE** html>

<**html** xmlns=*"http://www.w3.org/1999/xhtml"*

xmlns:h=*"http://xmlns.jcp.org/jsf/html"*

xmlns:p=*"http://xmlns.jcp.org/jsf/passthrough"*

xmlns:f=*"http://xmlns.jcp.org/jsf/core"*>

<**h:head**>

<**title**>Product Table </**title**>

<**f:metadata**>

<**f:event** listener=*"#{productController.loadProducts()}"*

type=*"preRenderView"* />

</**f:metadata**>

</**h:head**>

<**h:body**>

<**h:form**>

<**h:commandButton** value=*"Edit Data"* action=*"hello"* />

<**br** />

<**br** />

<**h:dataTable** value=*"#{productController.products}"* var=*"p"*

headerClass=*"table-header"* rowClasses=*"table-odd-row,table-even-row"*>

<**h:column**>

<**f:facet** name=*"header"*>Product</**f:facet**>

#{p.prodid}

#{p.proddesc}

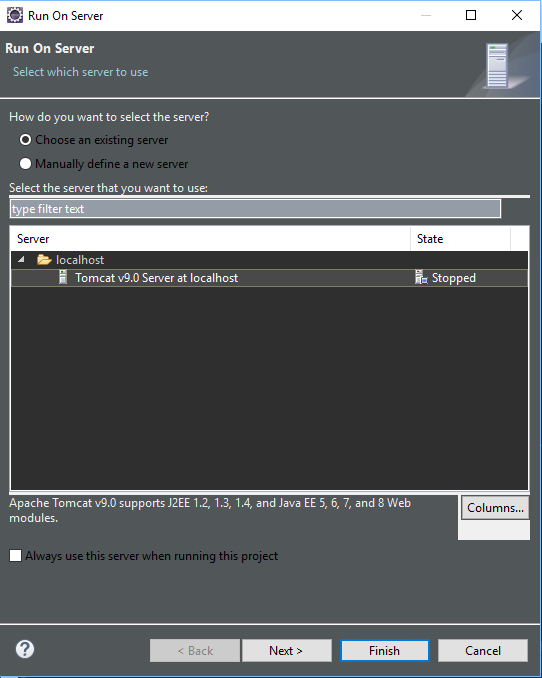
</**h:column**>

</**h:dataTable**>

</**h:form**>

</**h:body**>

</**html**>



After that go to elloOutput.xhtml and click run