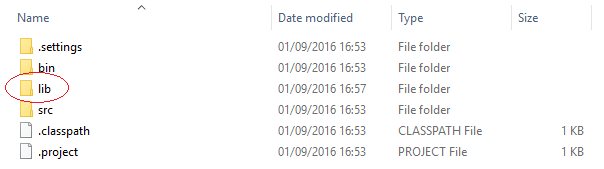
# Data Centric RAD

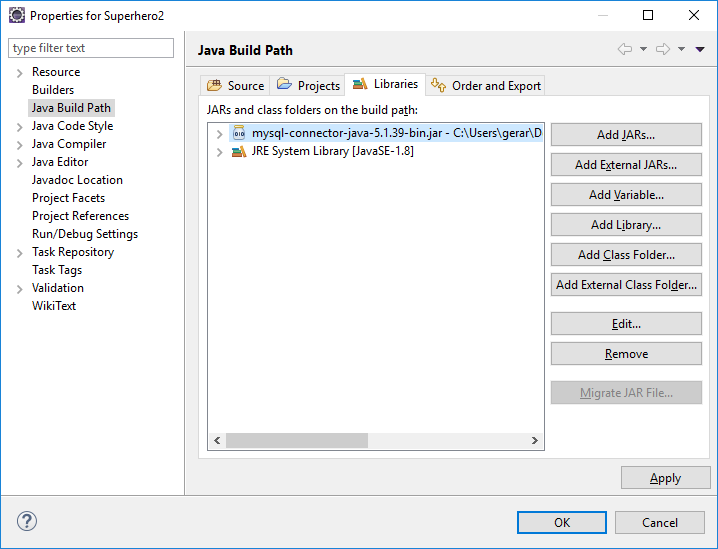
## Lab 8 JDBC PreparedStatement

### Setting up a JDBC Project

* Open Eclipse Java EE IDE for Web Developers.
* Select *File/New/Other…*
* When the dialog box opens, select *Java Project*.
* Give the Project a *Name* and press *Finish*.
* In the Windows File Explorer, navigate to the project you just created and create a folder called *lib*, so that your project now looks as follows:



* Download the MySQL driver from <https://dev.mysql.com/downloads/connector/j/>.
* After unzipping the file, copy mysql-connector-java-5.1.39-bin.jar to the *lib* folder previously created.
* In Eclipse, select the project you just created and then select *Project/Properties*.
* Click on *Java Build Path*, then the *Libraries* tab, then press the *Add External JARs…*
* Select the mysql-connector-java-5.1.39-bin.jar file in the lib directory of the project and press *Open.*
* When the mysql-connector-java-5.1.39-bin.jar appears on the build path (see below) press OK.



* In Eclipse, click on the *src* folder and select *File/New/Class*.
* Enter a Package name, and a Class name. Press *Finish*.
* At the top of the Class just created type the following:

**import** java.sql.Connection;

**import** java.sql.Statement;

**import** java.sql.ResultSet;

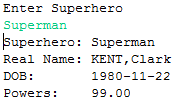
**import** java.sql.SQLException;

If there is no error saying the import cannot be resolved, this means the project is setup correctly.

* Start WAMP Server.
* To run each exercise below, select the .java file and select *Run/Run As/Run on Server* in Eclipse.

### Part 1 Get Superhero Details

* Get superheroes\_wk8.sql from Moodle.
* Import it into MySQL as described in Lab 1.
* Using the superheroes database, write a JDBC application that asks the user to enter the name of a Superhero, if a superhero that exists in the database is entered, the details of that superhero are displayed.



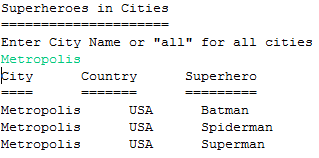
If the superhero entered does not exist in the database an error message is displayed informing the user of this.



The data should be retrieved from the database using a PreparedStatement.

### Part 2 Get Details of Superheroes in Cities

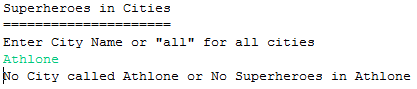
* Using the superheroes database, write a JDBC application that asks the user to enter the name of a City. The city name, the name of the country the city is in as well as the names of all superheroes in that city should be printed.



If the user enters *All* city name and country of all cities with superheroes are displayed.



If the user enters a non-existent city, or one in which there are no superheroes, the user should be informed of this.

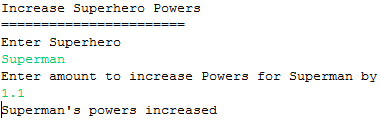


The data should be retrieved from the database using a PreparedStatement.

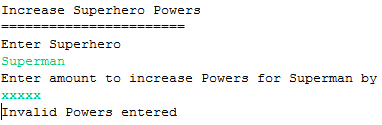
### Part 3 Increase Superhero Powers

* Using the superheroes database, write a JDBC application that asks the user to enter the name of a Superhero and the amount to increase his/her powers by.

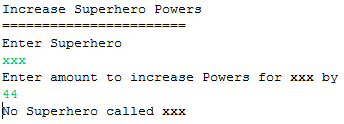
If the superhero entered exists in the database, and the powers entered by the user are in a valid format, the superhero’s powers should be increased by that amount.



If the user enters an invalid amount to increase the powers by he/she should be informed of this.



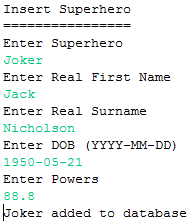
If the user enters a non-existent superhero he/she should be informed of this.



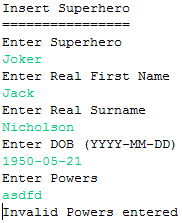
The data should be sent to the database using a PreparedStatement.

### Part 4 Add New Superhero

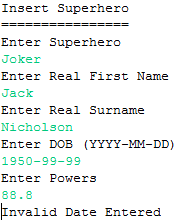
* Using the superheroes database, write a JDBC application that allows the user to enter a new Superhero.



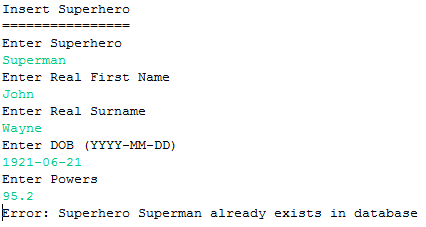
If the user enters invalid powers he/she should be informed of this



If the user enters an invalid date he/she should be informed of this



If the user enters an existing superhero he/she should be informed of this



The data should be sent to the database using a PreparedStatement.