CS 4513 – Fall 2017
Instructions for using JDBC on Oracle 12c

Step 1. Download JDBC driver ojdbc7.jar from the link:

http://www.oracle.com/technetwork/database/features/jdbc/default-2280470.html

Step 2. Write a *Java* program to use *Java JDBC* to connect to the Oracle database. Add *ojdbc7.jar* to the Java build path. If you are using eclipse, this can be done by right click *project name -> Properties -> Java Build Path -> Libraries -> Add External JARS* and select *ojdbc7.jar*.

Step 2.1. Import the java.sql package.

```
import java.sql.*;
```

Setp 2.2. Load a database driver.

```
try {
        Class.forName("oracle.jdbc.OracleDriver");

}
catch(Exception x){
        System.out.println( "Unable to load the driver class!" );
}
```

Step 2.3. Create an Oracle JDBC Connection. Replace the LoginName and Password in the statement below with your Oracle SQL Developer login name and password, respectively.

```
try{
Connection dbConnection=DriverManager.getConnection
("jdbc:oracle:thin:@//oracle.cs.ou.edu:1521/pdborcl.cs.ou.edu","loginName","Password")
}
catch( SQLException x ){
        System.out.println( "Couldn't get connection!" );
}
```

Step 2.4. Create an Oracle Statement object.

To execute SQL statements, you need to instantiate a Statement object from your connection object by using the createStatement() method.

```
Statement stmt = dbConnection.createStatement();
```

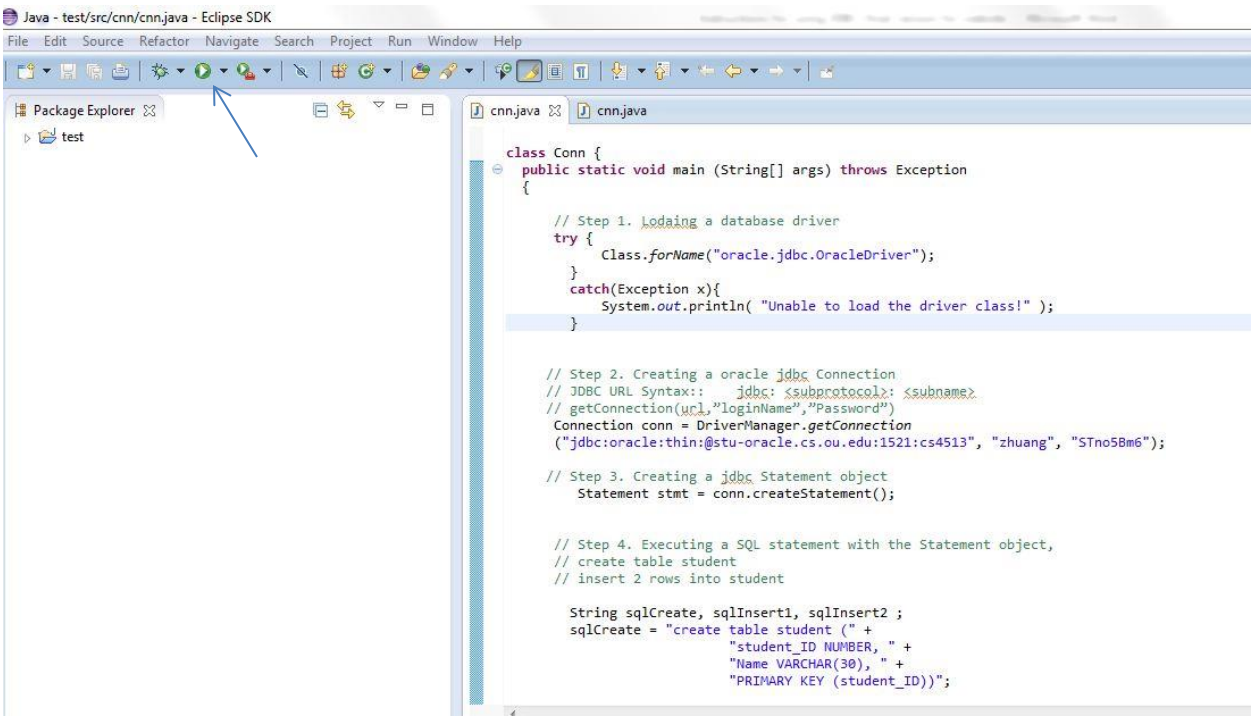
Step 2.5. Execute a SQL statement with the Statement object.

An example of a Java program using JDBC connection to perform database operations on Oracle 12c

```
// A sample Java program using JDBC to connect to an Oracle database,
// create a table student, insert 2 rows into student and
// display the IDs and names of all students
package testjdbc;
import java.sql.*;

class Conn {
public static void main(String[] args) {
        // Step 1. Loading a database driver
      String sourceURL = "jdbc:oracle:thin:@//oracle.cs.ou.edu:1521/pdborcl.cs.ou.edu";
      try {
                        Class.forName("oracle.jdbc.OracleDriver");
      }
        catch(Exception x){
                        System.out.println( "Unable to load the driver class!" );
        }

      // Step 2. Creating an Oracle JDBC Connection.  The following example assumes
      // that the login name is smith1234 and the password is johnsmith
      try{
        Connection conn = DriverManager.getConnection
        (sourceURL," smith1234 ", " johnsmith");

        // Step 3. Creating a JDBC Statement object
        Statement stmt = conn.createStatement();

        // Step 4. Executing the SQL statements with the Statement object

        // create table student
```

```
        // insert 2 rows into student
        String sqlCreate, sqlInsert1, sqlInsert2 ;
        sqlCreate = "create table student (" +
                                                "student_ID NUMBER, " +
                                                "Name VARCHAR(30), " +
                                                "PRIMARY KEY (student_ID))";

        sqlInsert1 = "insert into student values (100, 'Alice')";
        sqlInsert2 = "insert into student values (200, 'Bob')";

        try {
                stmt.executeUpdate(sqlCreate);
                stmt.executeUpdate(sqlInsert1);
                stmt.executeUpdate(sqlInsert2);
                } catch(SQLException ex) {
                        System.err.println("SQLException: " + ex.getMessage());
                }

       // display the IDs and names column of all students using the JDBC resultSet

      ResultSet rset = stmt.executeQuery("select * from student");
      System.out.println("Student ID     Student Name");
      while (rset.next())
          System.out.println (rset.getString(1) + "       " + rset.getString(2));
      conn.close();
      }
      catch(Exception e)
      {
        System.out.println (e.getMessage());
        System.out.println ("Exception occurred in executing the statement");
      }
  }
}
```

Step 3. To compile and run the program in *eclipse*, just press the RUN button pointed to by the arrow below.



References:

Some useful references for using JDBC:

Chapter 5 "Advanced SQL" in the textbook;

http://docs.oracle.com/javase/tutorial/jdbc/basics/index.html