# Department of Computing

**SE312: Software Construction**

**Class: BESE 6AB**

# Lab 06: Introduction to Java &JDBC

**CLO3: Develop applications and tools using various frameworks**

**Date: March 21, 2018**

# Time: Wednesday (9:00 – 12:00)

# Instructor: Zain ul Hassan Khan

# 

# Lab 6: Introduction to JAVA

**Introduction**

The Java Development Kit (JDK) is an implementation of either one of the Java SE, Java EE or Java ME platforms released by Oracle Corporation in the form of a binary product aimed at Java developers on Solaris, Linux, Mac OS X or Windows. Since the introduction of the Java platform, it has been by far the most widely used Software Development Kit (SDK)

**Objectives**

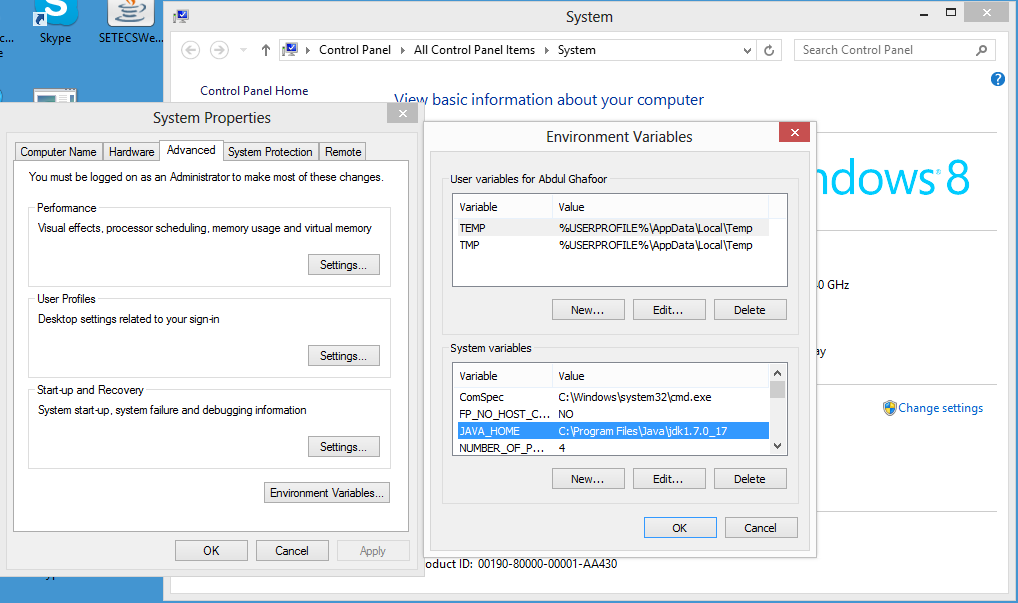
* How to install Java Development Kit and setting JAVA\_HOME as an Environment Variable
* Understanding basic structure of simple java program and its compilation/running process
* Understanding of java built in class and function invocation
* Understanding JDBC configuration and MySQL database connectivity with Java through JDBC.

**Tools/Software Requirement**

* Eclipse
* Netbeans

**Description**

* If JDK is already installed on your system then you can find it in <WindowsInstalledDir>:\Program Files\Java
* If it is not installed then download JDK from Internet
* <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
* Double click on downloaded exe and it will start wizard for JDK and runtime installation (jre)
* Right-click the My Computer icon on your desktop and select Properties.
* Click the Advanced tab. Click the Environment Variables button. Under System Variables, click New.
* Enter the variable name as JAVA\_HOME.
* Enter the variable value as the installation path for the Java Development Kit (java installed directory).
* Click OK.
* Select path and click Edit
* Add ;%JAVA\_HOME%\bin;%JAVA\_HOME%\lib
* Click Apply Changes.



Now open Eclipse / Netbeans software and create a new project and add a java HelloWorld class to your src folder in your project. And write the following code;

public class HelloWorld

{

public static void main(String arg[])

{

Scanner input =new Scanner(System.in);

String name = input.nextLine();

System.out.println("Welocme"+name+" to the JAVA world!");

}

}

Now Run this file.

**Lab Task – Part 1**

Write a java program that takes input of student names,Registration Number and their CGPA in a dimensional array. Assume there are 6 students in the class. [1 pt]

1. Check for valid CGPAs and duplicate entries (unique registration number) in your code. [2 pt]
2. Once the input is complete, print the complete data along with the min, max and average CGPA values for the class. [2 pt]
3. Print the names of the students who have obtained less than the average CGPA. [2 pt]
4. Now ask the user to type any name of student or registration number. Your program should now print whether the student data is available or not and if available what is the student CGPA and his rank in the class e.g., 2 out of 10. [3 pt]

**Lab Task- Part 2**

**Description**

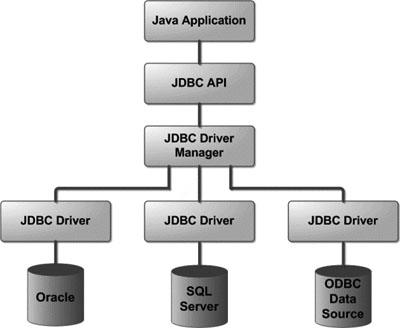
The JDBC API supports both two-tier and three-tier processing models for database access but in general JDBC Architecture consists of two layers:

* JDBC API: This provides the application-to-JDBC Manager connection.
* JDBC Driver API: This supports the JDBC Manager-to-Driver Connection.

The JDBC API uses a driver manager and database-specific drivers to provide transparent connectivity to heterogeneous databases.

The JDBC driver manager ensures that the correct driver is used to access each data source. The driver manager is capable of supporting multiple concurrent drivers connected to multiple heterogeneous databases.

Following is the architectural diagram, which shows the location of the driver manager with respect to the JDBC drivers and the Java application:



**Common JDBC Components**

The JDBC API provides the following interfaces and classes:

* **DriverManager:** This class manages a list of database drivers. Matches connection requests from the java application with the proper database driver using communication subprotocol. The first driver that recognizes a certain subprotocol under JDBC will be used to establish a database Connection.
* **Driver:** This interface handles the communications with the database server. You will interact directly with Driver objects very rarely. Instead, you use DriverManager objects, which manage objects of this type. It also abstracts the details associated with working with Driver objects
* **Connection:** This interface with all methods for contacting a database. The connection object represents communication context, i.e., all communication with database is through connection object only.
* **Statement:** You use objects created from this interface to submit the SQL statements to the database. Some derived interfaces accept parameters in addition to executing stored procedures.
* **ResultSet:** These objects hold data retrieved from a database after you execute an SQL query using Statement objects. It acts as an iterator to allow you to move through its data.
* **SQLException:** This class handles any errors that occur in a database application.

For examples: <http://www.tutorialspoint.com/jdbc/jdbc-create-database.htm>

**Lab Task**

Create a MySQL database “University” which has a table “Student” having columns “ID”, “RegNo”, “Name”, “Class”, “Section”, “Contact” and “Address”. Add 5 records to the table.

Now, create a connection to this database in java using JDBC. Display a menu for user to perform the following activities:

* Display all the records
* Delete a specific record requested by user
* Search a specific record entered by user

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Activity** | **Minimum** | **Maximum** |
| Part 1 | 0 | 4 |
| Part 2 | 0 | 2+2+2 |
| **Total** | **0** | **10** |

## Deliverables

* Each submission is individual with the following composition:
  + Source Code
  + Documentation (Comments)
  + Link to the public repository preferably on GitHub
* Convert your submission files into a zip folder and name it as given below, finally upload the zip folder to LMS.
  + Name – Registration No. – Section

## Grade Criteria

This lab is graded. Min marks: 0. Max marks: 10.

**Links for Learning:**