CSI 2132 Lab#4

- JDBC

Presented by: 03 Feb 2020

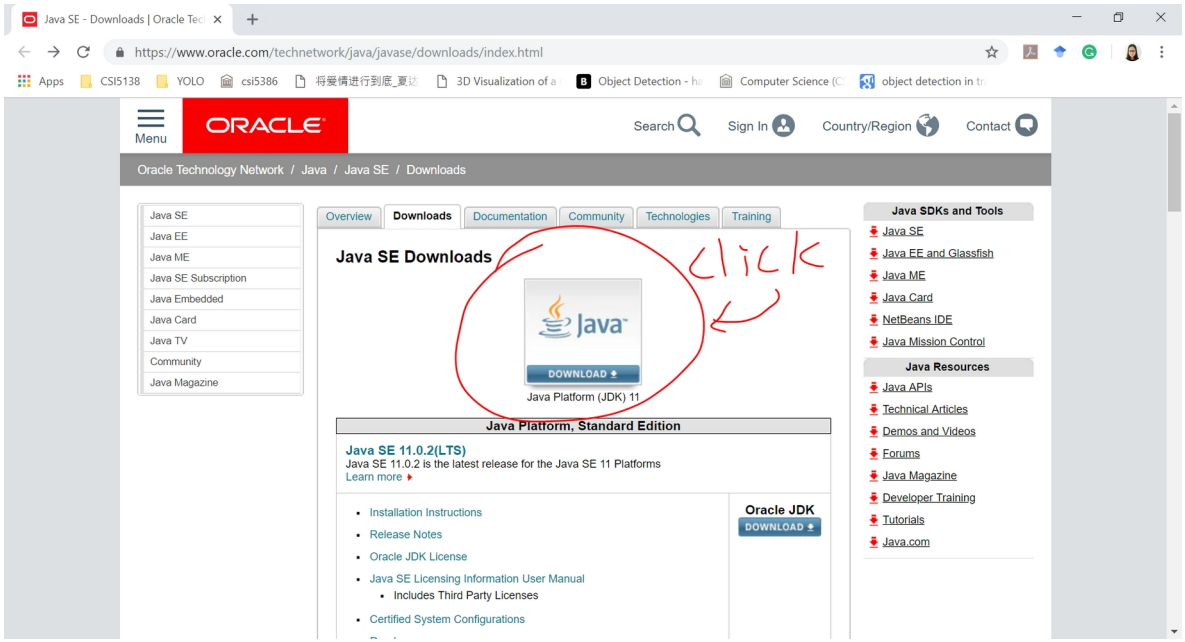
- JAVA
- JDK

Using 'java', 'javac', 'java -version' to check if you have installed

Install JDK(Java Development Kit)

- Download JDK

https://www.oracle.com/technetwork/java/javase/downloads/index.html

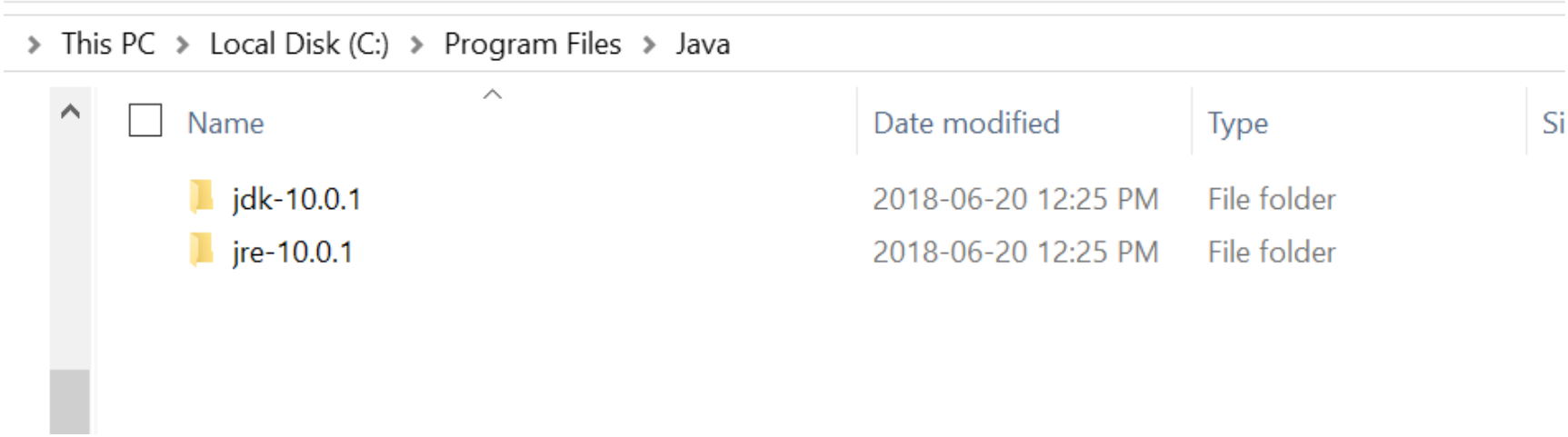


Install JDK(Java Development Kit)

- Install Java (JDK)

  Just click next and use default settings

  both JDK and JRE will be installed

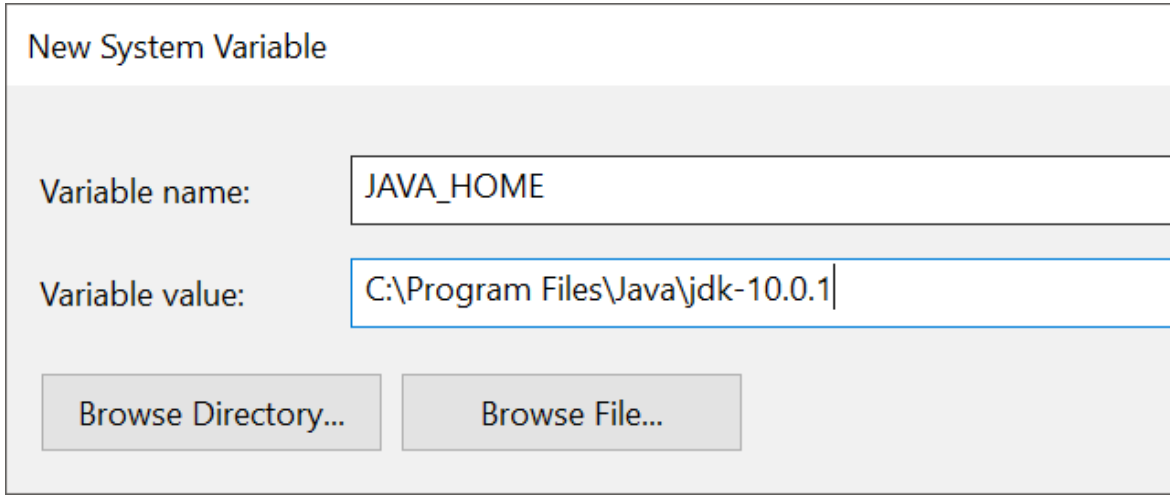


Install JDK(Java Development Kit)

- Configuring Java environment variable

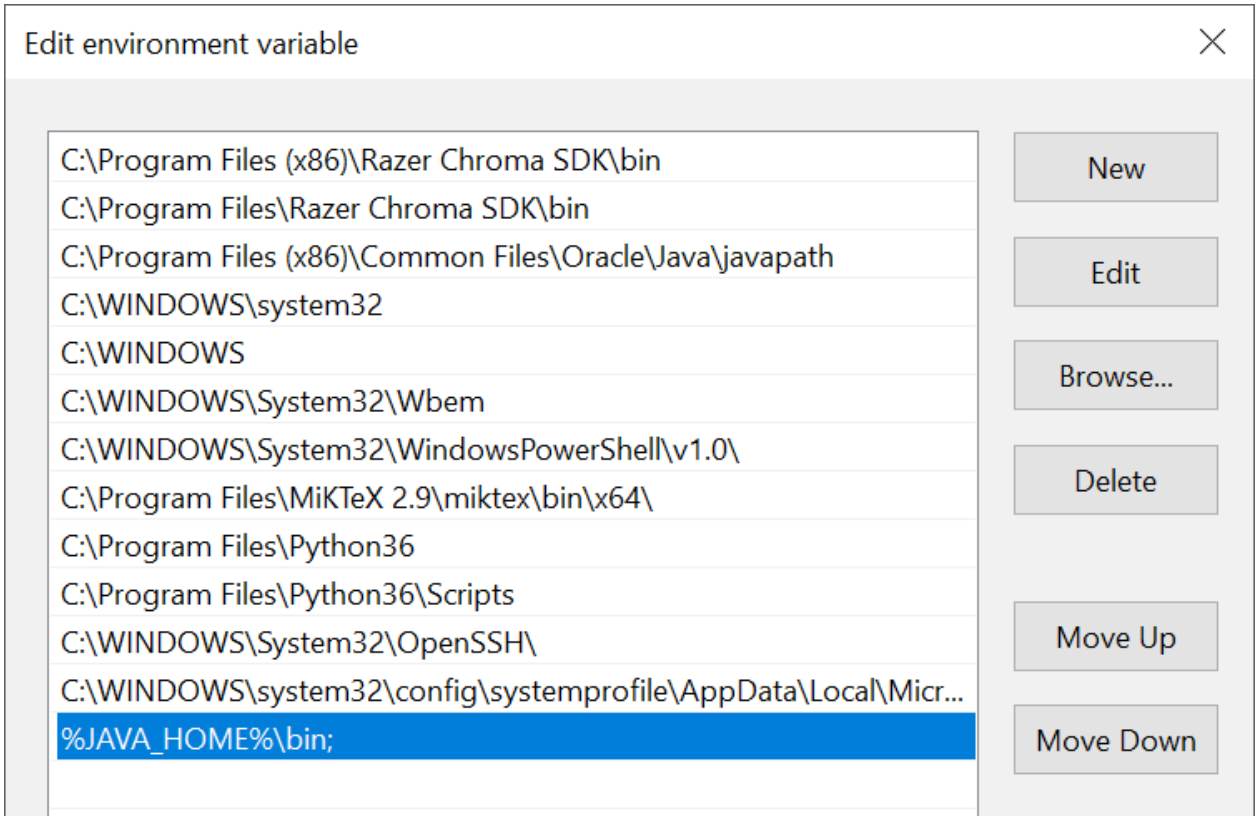
  Right click 'this pc' -> properties -> advanced system settings -> Environment Variables

  Create a new environment variable called **JAVA_HOME**.



Install JDK(Java Development Kit)

- Add **'%JAVA_HOME%\bin;'** to path



Let's try to write 1st JAVA program

test1.java

```
public class test1 {
    public static void main(String args[]) {
        System.out.println("Welcome to JAVA World!");
    }
}
```

```
C:\Users\Zhijun Hou\Desktop\CSI2132\week 4\lab#4>javac test1.java

C:\Users\Zhijun Hou\Desktop\CSI2132\week 4\lab#4>java test1
Welcome to JAVA World!
```

Outline

- What is JDBC?
- JDBC Driver (download and use)
- Java programming with JDBC
- Dynamic SQL queries with Java

What is JDBC

- Java Database Connectivity (JDBC)
- It is an API(Application programming interface) by Sun Microsystems to allow Java programmers to access SQL databases
- Available since JDK 1.1
- JDBC is an API not a library. It needs to be implemented (as drivers) for a particular DB. i.e. PostgreSQL and MySQL have different JDBC drivers
- In this course we use PostgreSQL so we download PostgreSQL JDBC driver

Download JDBC Driver

- Download JDBC from:
  - https://jdbc.postgresql.org/download.html

- *JDBC4 PostgreSQL driver– version 9.0-801*

Current Version *42.2.5*

This is the current version of the driver. Unless you have unusual requirements (running old applications or JVMs), this is the driver you should be using. It supports PostgreSQL 8.2 or newer and requires Java 6 or newer. It contains support for SSL and the javax.sql package.

- If you are using Java 8 or newer then you should use the JDBC 4.2 version.
- If you are using Java 7 then you should use the JDBC 4.1 version.
- If you are using Java 6 then you should use the JDBC 4.0 version.
- If you are using a Java version older than 6 then you will need to use a JDBC3 version of the driver, which will by necessity not be current, found in Other Versions.

PostgreSQL JDBC 4.2 Driver, 42.2.5

PostgreSQL JDBC 4.1 Driver, 42.2.5.jre7

PostgreSQL JDBC 4.0 Driver, 42.2.5.jre6

JDBC Enabled Project in eclipse

1. Create a new Java Project in eclipse (jdbc)
2. From the Project Properties, click on
   Java Build Path > Libraries > Add External JARs
3. Select the downloaded JAR file
4. Create a new package in your project (code)
5. Create a new class inside the package with a static main method (connection.java)
6. Write a try – catch structure inside the main method with a generic exception handler

JDBC Coding in JAVA

1. Import the JDBC driver
2. Load the driver
3. Connect to a Database
4. Issue a Query and process the result

JDBC Coding in JAVA

1. Import the JDBC driver
   - Import java.sql.*;
   - *It is NOT appropriate to import org.postgresql directly*
   - *Remember the import lines go after the package line*
2. Load the driver
3. Connect to a Database
4. Issue a Query and process the result

JDBC Coding in JAVA

- Import the JDBC driver
- Load the driver
  - Class.forName("org.postgresql.Driver");
  - *You can check that this class actually exist under Referenced Libraries > postgresql-9.0-801.jdbc4 > org.postgresql > Driver.class*
- Connect to a Database
- Issue a Query and process the result

JDBC Coding in JAVA

- Import the JDBC driver
- Load the driver
- Connect to a Database
  - *Connection db = DriverManager.getConnection(url, username, password);*
- URL is in the form of:
  - jdbc:postgresql:*database*
  - jdbc:postgresql://*host*/*database*
  - jdbc:postgresql://*host*:*port*/*database*
  - jdbc:postgresql://web0.site.uottawa.ca:15432/svale054
- Username: your SITE username (svale054)
- Password: your SITE password (XXXXXX)
- Issue a Query and process the result

JDBC Coding in JAVA

- Import the JDBC driver
- Load the driver
- Connect to a Database
- Issue a Query and process the result

```
Statement st = db.createStatement();
ResultSet rs = st.executeQuery("SELECT * FROM laboratories.artist");
while (rs.next()) {
System.out.print("Column 1 returned: ");
System.out.println(rs.getString(1));
}
rs.close();
st.close();
```

Example

Write a Java program (FirstExcercise.java) that connects to our own database and retrieves the name and birthday of all artists. Print the result as a 2D table using System.out.print

Dynamic Queries

```
String field = “aname, Style";

String cond = “aname";

String table = “laboratories.artist";

String value = “Caravaggio";

Statement st = db.createStatement();

ResultSet rs = st.executeQuery("SELECT " + field + " FROM " + table +
" WHERE " + cond + " = '" + value + "';");
```

Dynamic Queries

- To get number of columns returned by the query:

```
ResultSetMetaData rsMetaData = rs.getMetaData();
int numberOfColumns = rsMetaData.getColumnCount();
```

Exercises

- Write Java code (SecondExcercise.java) that returns those fields of table Artist that are in an array named fields:
  - String[] fields = {"AName", "Style", ...};
- Allow your program to retrieve information from more than one artist (hint: use keyword IN).
- Try changing the fields array and recompile and run your code. It should work for all valid fields