

Lab 0: Infrastructure Setup

Instructor: Vivek Sarkar. Co-Instructor: Shams Imam

Course Wiki: <http://comp322.rice.edu>

Staff Email: comp322-staff@mailman.rice.edu

Goals for this lab

- edX Setup
- Piazza Setup
- Java 8 installation
- Maven installation
- IntelliJ installation

Important tips and links

NOTE: The instructions below are written for Mac OS and Linux computers, but should be easily adaptable to Windows with minor changes. For example, you may need to use `\` instead of `/` in some commands.

Note that all commands below are CaSe-SeNsItIvE. For example, be sure to use “S16” instead of “s16”.

edX site: <https://edge.edx.org/courses/RiceX/COMP322/1T2014R>

Piazza site: <https://piazza.com/rice/spring2016/comp322/home>

Java 8 Download: <https://jdk8.java.net/download.html>

Maven Download: <http://maven.apache.org/download.cgi>

IntelliJ IDEA: <http://www.jetbrains.com/idea/download/>

1 edX Setup

We will use COMP 322's edX site, <https://edge.edx.org/courses/RiceX/COMP322/1T2014R>, for hosting videos and quizzes. Please only register for COMP 322 on edX with your email address of the form, *your-netid@rice.edu*.

If you can access the above edX site, you are done and you can move on to the next section. If not, please send email to comp322-staff@mailman.rice.edu with a request to add your email address to the edX enrollment list for this class.

2 Piazza Setup

We will use COMP 322's Piazza site, <https://piazza.com/rice/spring2016/comp322/home>, for all discussions and Q&A. Please only register on this site with your email address of the form, *your-netid@rice.edu*.

If you can access the Q&A tab in this Piazza site, you are done and you can move on to the next section. If not, please send email to comp322-staff@mailman.rice.edu with a request to add your email address to the Piazza enrollment list for this class.

3 Java 8 Setup

You may already have Java 8 installed on your computer from COMP 215. To check whether you have Java 8 installed on your machine, go to the command line and type the following `java -version`. If you see something as follows:

```
$ java -version
java version "1.8.0_72-ea"
Java(TM) SE Runtime Environment (build 1.8.0_72-ea-b05)
Java HotSpot(TM) 64-Bit Server VM (build 25.72-b05, mixed mode)
```

where the Java version shows 1.8.*, you already have Java 8 installed on your machine. *If you do have Java 8 already installed on your machine, please skip to section 4.*

If you do not already have it installed, you will need a Java 8 installation on your machine and have your `JAVA_HOME` and `PATH` point to the new installation. Java 8 can be downloaded and installed from the [Oracle website](#).

For example, I have the following on my Mac machine's `.bash_profile`:

```
export JAVA_HOME=/Library/Java/JavaVirtualMachines/jdk1.8.0_72.jdk/Contents/Home
export PATH=$JAVA_HOME/bin:$PATH
```

On Windows, the environment variables have to be set up differently. Please refer to the [stackoverflow question](#) to see how it can be done.

4 Maven Setup

Maven is a build automation tool used primarily for Java projects. Projects using Maven or other build systems are easiest to use as they simplify compiling, building, and testing the project. In addition, major IDEs like IntelliJ and Eclipse have excellent support via maven plugins which simplifies the development process. Dependency management is one of the areas where Maven excels; for our purposes, this means that we will save a lot of effort in configuring the projects to set up dependencies on HJlib, JUnit, and other jars. Hence, for the labs and assignments in COMP 322, we will distribute maven project templates to be used by students to complete their work.

Maven is a Java tool, so you must have Java installed in order to proceed. Next, follow the [instructions on the maven site](#) to install maven. Please remember to install a version that is 3.3.3 or later. Detailed instructions are also available on the COMP 322 wiki at <https://wiki.rice.edu/confluence/display/PARPROG/Using+Maven+for+HJlib+projects>. During installation have your `M2_HOME`, `MAVEN_HOME` and `PATH` point to the new installation. For example, I have the following on my Mac machine's `.bash_profile`:

```
export M2_HOME=/Users/shamsimam/dev/apache-maven-3.3.3
```

```
export MAVEN_HOME=/Users/shamsimam/dev/apache-maven-3.3.3
export PATH=$PATH:$MAVEN_HOME/bin
```

Once installed, open a new command prompt and run `mvn --version` to verify that it is correctly installed. If you see something as follows:

```
$ mvn --version
Apache Maven 3.3.3 (7994120775791599e205a5524ec3e0dfe41d4a06; 2015-04-22T06:57:37-05:00)
Maven home: /Users/shamsimam/dev/apache-maven-3.3.3
Java version: 1.8.0_72-ea, vendor: Oracle Corporation
...
```

Maven has been successfully installed on your machine.

If you have unzipped the contents of the Maven zip file and notice an output similar to the following:

```
$ mvn --version
-bash: mvn: command not found
```

it means that your `PATH` variable has not been configured correctly.

5 IntelliJ Installation

An IDE is not strictly required for this course since Java programs can also be written using a text editor and then compiled using the command-line. However, we strongly recommend using an IDE as it simplifies writing, compiling, running, and debugging your programs. A good IDE gives you error warnings, code completion and navigation, refactoring, syntax highlights, etc. We recommend using the IntelliJ IDE to do the Java development in the labs and assignments for this course. A free version of IntelliJ (Community Edition) IDE can be downloaded and installed from the [Jetbrains website](#).

6 Use of JUnit in COMP 322

JUnit is a unit testing framework for the Java programming language. As in COMP 215, we will use JUnit for all labs and programming assignments. You will be provided basic JUnit tests for your labs and assignments. Note that, in Maven projects, there are separate directories for the source code and unit tests. Since Maven will handle our dependencies, it will automatically download the necessary JUnit jars. We will be using version 4.7 of JUnit, this can be determined by checking the version in the `pom.xml` file of each individual Maven project.