Lab 0: Infrastructure Setup
Instructor: Mackale Joyner. Co-Instructor: Zoran Budimlić

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

Goals for this lab

- Piazza Setup

- Java 8 installation

- Maven installation

- IntelliJ installation

- Subversion integration

- Introduction to HJlib Installation and Setup

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 "S19" instead of "s19".

Piazza site: https://piazza.com/rice/spring2019/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   Piazza Setup

We will use COMP 322's Piazza site, https://piazza.com/rice/spring2019/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 the instructors with a request to add your email address to the Piazza enrollment list for this class.

2   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 3.*

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.

3   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; 2019-01-05T06:57:37-05:00)
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.

4   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.

5   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.

6   Subversion Setup

Subversion is a software versioning and revision control system. This allows you to recover older versions of your data or examine the history of how your data changed. During COMP 322 we will use your private repository to distribute code to you. You can always examine the most recent contents of your svn repository by visiting `https://svn.rice.edu/r/comp322/turnin/S19/`*your-netid*. *If you are unable to access the above URL, please send email to helpdesk@rice.edu cc'ing comp322-staff@mailman.rice.edu and requesting that they fix your access. After that, you can ignore this section for now (till you get access) and move on to the next section.*

We have a strict naming convention for folders in the svn repository— "`hw_1`", "`hw_2`'', ... for homeworks and "`lab_1`", "`lab_2`", ... for labs.

There are primarily three SVN commands you will need to be familiar with for COMP 322: `checkout`, `commit`, and `add`. An SVN `checkout` downloads files from the SVN cloud to your local laptop, like downloading the provided source code template for labs and homeworks from your private SVN repository. An SVN `commit` uploads files from a checked-out version of the SVN repo on your local laptop back to the SVN cloud. It will only upload the changes you have made to those files. You can look back through your commits to see the changes you have made over time. `commit` updates the cloud SVN repo with changes you made to your local copy. If you create a new file inside a local, checked-out SVN repo, you will first have to SVN `add` it to inform SVN that there is a new file it should start uploading with your commits.

There are a few ways you can interact with your subversion repository: use svn with IntelliJ projects, use the Subclipse plugin for Eclipse, use a standalone SVN client like Tortoise SVN, or use the command line.

7   Lab 1 Exercises

Visit the url at `https://svn.rice.edu/r/comp322/turnin/S19/`*your-netid*`/lab_1` to ensure that the files for this lab are available. The directory structure for the lab should look like the following:

```
pom.xml
src
    main
        java
            edu
                rice
                    comp322
                        HelloWorldError.java
                        ReciprocalArraySum.java
    test
        java
            edu
                rice
                    comp322
                        Lab1CorrectnessTest.java
```

Note that you will need to have Java 8, Maven, and IntelliJ installed for this step. You can download the
lab_1 project on your machine through either the command line or using IntelliJ's SVN support. The two
sections below contain instructions on both approaches, so you only need to follow one.

7.1 Using IntelliJ to checkout your SVN repo

Checking out your Lab 1 SVN folder from the remote SVN server is straightforward in IntelliJ. You can
start from either the "Check out from Version Control" option on the welcome screen:

or use the "Checkout from Version Control" option accessible under the top menu's VCS tab:
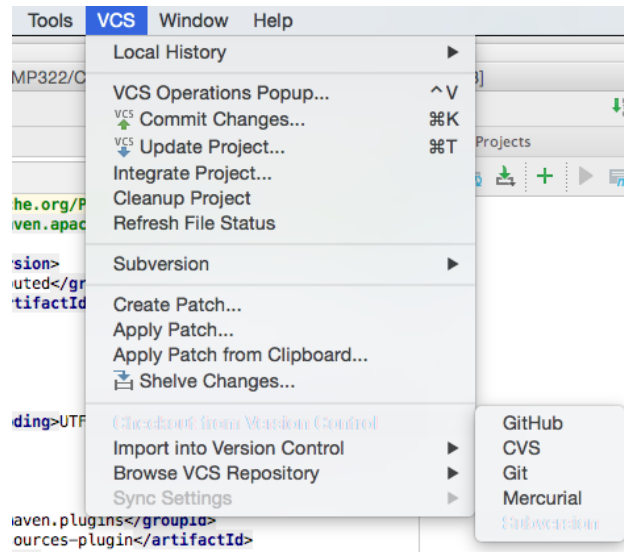
Use the plus button in the pop-up window to add the following SVN folder:

`https://svn.rice.edu/r/comp322/turnin/S19/NETID/lab_1`

with NETID replaced by your Net ID. Click the Checkout button to start downloading this repo.
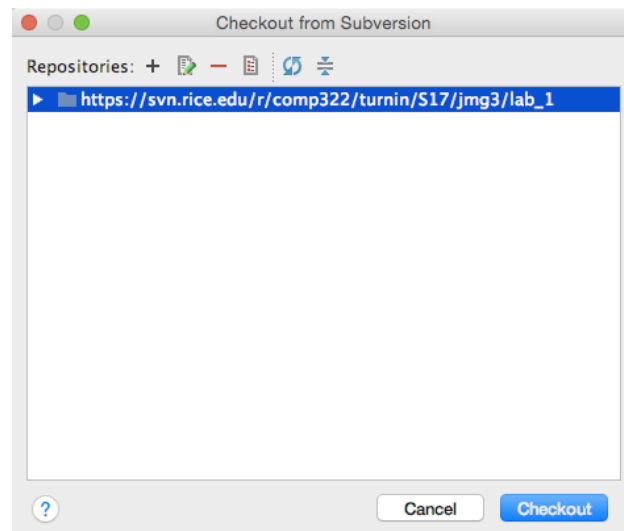
The next prompt will query you for the folder name to checkout the local copy of the SVN folder to. The choice of folder is up to you. Once you have made a choice, click OK.
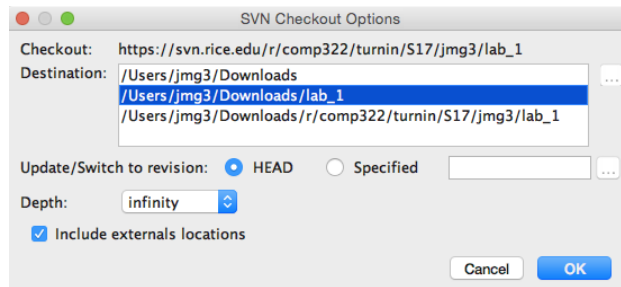
IntelliJ will likely then ask you to choose the SVN format to use for the local copy of the remote folder. We recommend you simply pick the latest option available and hit OK:
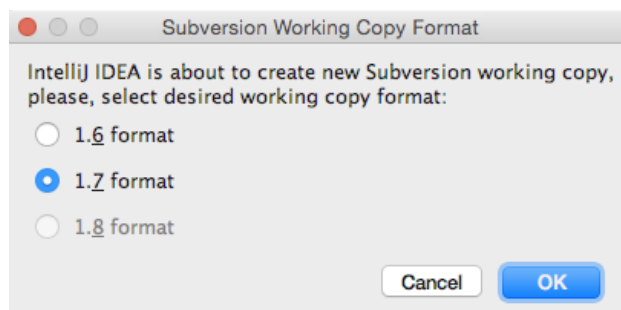
IntelliJ will now ask if you would like to open the newly checked-out project. Hit Yes:
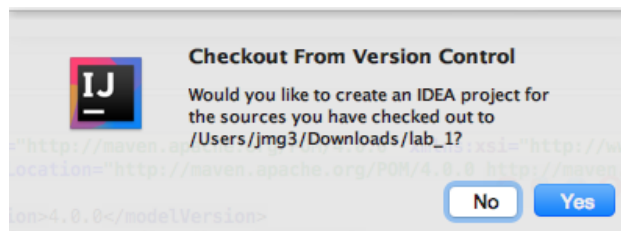
On the next screen, you may be asked if you would like to import this project as a Maven project (your IDE may also skip this step and auto-detect it as a Maven project, in which case you can skip ahead):
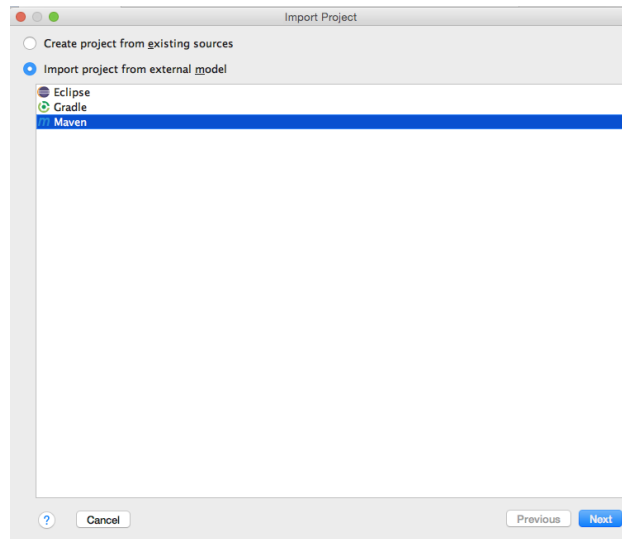
Finally, IntelliJ should open your project and may ask you if you would "like to schedule the following file for addition to Subversion?". Select No. You are now ready to get started!

7.2 Using the command line to checkout your SVN repo

Note that you do not need to follow these instructions if you have already checked the project out through IntelliJ. These instructions are for those who would prefer to work on the command line. The command-line SVN client can be installed by following the instructions at Apache Subversion Binary Packages. Once installed, you can verify that the command works by running the following command:

```
$ svn --version
  svn, version 1.8.11 (r1643975)
  compiled Jan  5 2019, 13:33:40 on x86_64-apple-darwin13.4.0
  ...
```

To checkout your SVN repo for Lab 1 on the command line, navigate in a terminal to the directory you would like to checkout into and issue the following command:

```
$ svn checkout https://svn.rice.edu/r/comp322/turnin/S19/NETID/lab_1
```

You should see a lab_1 directory created in your working directory with the source code for this lab.