Assignment 1

Summary

Install an Open-Source Java SE JDK, Eclipse, Lombok, and Spring Tools, then run an ultra-easy Spring Boot Hello World.

This will be an exercise in problem solving for many of you, so spend some time and really try to figure it out on your own (maybe with Google!). You'll be glad you did! I will not be holding your hand in this course, and instead expect you to use all legit resources you have available!

Step 1 – The Latest Java JDK!

There may already be a Java SE JDK installed on your machine, but new versions are released all the time, and it really pays to be up to date. Most new Java releases patch security holes, but for whatever reason, Java doesn't usually uninstall the old, EXTREMELY prone to malicious attack, versions from your machine! Do yourself a favour – uninstall them manually every time you upgrade, and always, always, always upgrade!

Additionally, Oracle recently changed their licensing agreement for the formerly standard Java JDK such that they "own" the rights to some code produced with it, so we'll be switching to the recommended AdoptOpenJDK instead. Temurin is a version of Adopt Open JDK, an open-source rival to the official Oracle JDK, and is the officially recommended JDK version for use with Spring.

Therefore, uninstall any old versions of Java through Windows Add and Remove Programs (or other appropriate location depending on your OS). Having old versions around leaves you open to exploits from the bad guys.

Navigate to https://adoptium.net/, and download and run the OpenJDK **17** (LTS) Installer. (jdk-17.0.8.1+1 LTS Release as of this writing) Default installation options are fine. Now, reboot.

Worth saying… we all need to be on the same page with this one. **It's tough to mark your work if we are all on different setups** – and again, older Java versions are also extremely dangerous and full of exploits anyway! Uninstall them all, install **Temurin 17**, and reboot to be sure please!

Step 2 – Eclipse

Navigate to the following link: https://eclipse.org/downloads/ Click to download the latest Eclipse installer (2023-06 as of this writing). Execute the installer package and select the **Eclipse IDE for Enterprise Java and Web Developers** option. Emphasis on the **Enterprise Java** Developers edition – there are many possible versions of Eclipse, Eclipse for both regular and Enterprise Java, other entirely different languages, obviously, and even some spin-offs floating around.

Be sure you switch your Eclipse setting for your JVM, or the Java 17+ VM folder, to reflect your newly downloaded Eclipse Temurin JDK 17 version.

Also, take note of where your workspace is located! The defaults are only ok here - I prefer to store mine in directories I create for various semesters. Do **not** use a Cloud file storage system with Eclipse or your workspace like OneDrive or Google Drive or Dropbox! They're waaaaaaay too slow for our needs, and your rig will almost certainly crash - a lot!

You can have as many versions of Eclipse running on your computer as you like. An Eclipse modified for Android (like the old Android Studio) will not interfere with a separate Eclipse modified to become SpringSource Tool Suite, and, for that matter, Eclipse for Enterprise Java will not even interfere with a second Eclipse for Enterprise Java in a different directory. Multiple running servers like Tomcat, GlassFish, JBoss, or Jetty, on the other hand, **will definitely** interfere with each other quite readily, so be careful.

Launch Eclipse from the installer when directed, then right-click on the loaded and running icon on your Taskbar and opt to Pin to Taskbar if you're using Windows. This makes it nice and easy to load Eclipse from virtually anywhere, which we'll be doing in almost every class.

In your running Eclipse, select Hide from the top right. Now, close Eclipse.

Step 3 - MySQL

Navigate to http://dev.mysql.com/downloads/windows/installer/ and download the MySQL Installer (the smaller file size option is faster). The latest version at the time of writing is 8.0.34. There is only a 32-bit version of the installer, but not to worry as it will know what type of architecture you're running while it does its thing. This version of MySQL is free and includes an easy-to-use workbench for structure creation and manipulation – all in one package!

Note: no need to create an account here. Just jump to the bottom of the page and click "No thanks, start my download".

Run the downloaded file. Warning, this is a pretty complicated process, and even though the installer will automatically uninstall and upgrade or install any and all MySQL components for you, at least one or two things almost always go wrong. If it happens to you, try letting it get as far as it can before cancelling and running it again. You can also try finishing (going all the way to the end even with errors) what you can and then running it again, and of course, a handy reboot-then-try-again may help too.

Officially, you only require the 32 or 64-bit Server, though the corresponding 32 or 64-bit Workbench GUI is also likely to be very helpful. That said, feel free to install anything else you feel is interesting!

Enter your existing root password if you have one but be sure to change it to **1234**. Email yourself if you have to! I can't tell you how many times people have asked me how to reset their password because they've forgotten it, **and it is virtually impossible to reset!** As your machine will just be a development machine, there will not be anything on it that needs serious security anyway. Please do not be stubborn and insist on using your own password because it will be exposed in the code for all to see in anything you hand in or present to the class, and your code will crash on my machine while I am marking your assignments and exams. **System crashes usually result in significant lost marks!**

Step 4 - Lombok

Navigate to https://projectlombok.org/download to download and install the latest version of the Lombok Bean Management Framework. At time of writing the latest version is 1.18.28.

Lombok will allow us to create beans using only minimal code and may be a happy surprise when you first see it in action. It will look like it lives between the Java code and the complier, though in reality it just modifies our IDE to use its own Lombok libraries, in addition to any others we may specify, at compile time. Eclipse is kind enough to compile our code virtually every time we save, so we'll see Lombok in action in the drilldowns for methods of a bean class. Beware! The methods we may see Lombok generate do not necessarily exist as source code, except in annotation form, but will be compiled as though they exist when Lombok steps in to intercept the relevant annotations. When it works, it will look like it adds in getters, setters, toStrings, equals, constructors, and more for our bean properties by itself!

Execute the .jar you downloaded to start the installation. Not entirely sure why, but this may mean you need to navigate to your Downloads (or wherever you saved it) directory and run it manually. Browsers are often unhappy executing downloaded JARs directly, and for good reason! – it's a solid security thing.

Specify your installed Eclipse executable from the previous step. It will probably not be able to find it on its own if you put it in a special directory for our course – use the Specify Location wizard to point Lombok at the actual eclipse.exe file. Click Install/Update.

Note: we will still need to Maven in Lombok.jar for our projects to use it properly. We have only modified Eclipse's compilation settings to compile with Lombok in the correct order. All in good time…

Additional Note: I have on occasion needed to modify my eclipse.ini file in my Eclipse install directory to remove the absolute path reference to the Lombok JAR. Rather than some long and problematic path, possibly including spaces or special characters, the last line in my eclipse.ini file in my Eclipse directory only reads as follows to prevent an Eclipse load error:

```
-javaagent:lombok.jar
```

Step 5 – Spring Tools

Restart Eclipse by clicking on your new Taskbar icon!

In Eclipse, click Help → Eclipse Marketplace. This is a sort of app store for Eclipse, though more truthfully, it could be called a plugin repository. Technically, Eclipse is open source and therefore free, so store doesn't quite cut it, and though Windows and Macs are set up to run desktop apps, Eclipse is an app onto itself, so app doesn't really fit either.

For the interested, Eclipse is officially an IDE to make IDEs, and therefore is fully API'ed into bazillions of possible things one can do programmatically or through modular plugins. Many if not most of your favourite IDEs probably started as Eclipse at one point or another, including IntelliJ, SpringSource Tools Suite, and Android Studio. Friendly note to look around in here if you have a few minutes for other amazing plugins, though I can appreciate it's not always easy to know what's what just yet.

Spring Source Tools Suite has been packaged into a simple plugin we can now easily install here. Beside the magnifying glass for searching in the Eclipse Marketplace window, type **spring**. Install **Spring Tools 4** (aka Spring Tool Suite 4) v4.19.1.RELEASE at the time of writing.

This process may take a minute or two. Accept any licenses and (eventually) restart Eclipse when prompted – you should be able to see what Eclipse is doing in the bottom right-hand corner if it seems to be taking a while or you're ever unsure.

Step 6 – Spring Boot Hello World!

In Eclipse, click File → New → Other → Spring Boot → Spring Starter Project.

In the name field, type **A1YourName**, with your own first and last name, of course. Friendly note, I'll be using a naming convention for the remainder of the course for lectures, exercises, assignments, and exams. Exercise 1-2 is the exercise component of week one on the second day, for example, so would be named ex12_SomethingErOther. For the purposes of illustration, the lecture demo from the second day in week five might be lec52_SomethingErOther.

Do not use dashes or dots in project and package names! It will hopefully become very clear that dots are unbelievably confusing to the JDK, and let it be known dashes are also often converted to dots.

Set the type as Maven, the packaging as Jar and the language as Java. Change the Java Version to 17 if needed though! Friendly thought, if you see something other than 17 in Java Version, it likely means you need to go back to the previous step and uninstall then reinstall the JDK. Do that first.

Change the group field to ca.sheridancollege.<YourUserName>, where <YourUserName> is your Sheridan username or email alias. This step is important! Most group id's in Maven and elsewhere are reverse URLs, and as we're all at sheridancollege.ca, it should make a certain amount of sense. This does not mean you can automatically load YourUserName.sheridancollege.ca and have this thing run though. All in good time, and I hope you will deploy your Jars or Wars to the cloud of your choice in future semesters.

The Artifact ID is the name of this particular project at your URL, and should reflect your project name of A1YourName all by itself.

The Version is fine for now. The Description is fine for something as simple as HelloWorld at least too.

Change the Package field to ca.sheridancollege.<yourUserName> just like we did with the group ID. Click Next.

On the next page is a list of dropdowns and checkboxes. Believe it or not, it allows us to build in HUGE libraries of code as our project is created! By the end of the course, we'll want the following:

Developer Tools → Spring Boot DevTools

Developer Tools → Lombok

I/O → Java Mail Sender

SQL -> Spring Data JDBC

SQL → H2 Database

SQL -> MySQL Driver

Security → Spring Security

Template Engines → Thymeleaf

Web → Spring Web

Web -> Spring Session

Click the Make Default button when you have them all selected.

Note that this is an exhaustive list for our course, but we will not get to many of them until much later in the semester. Also, note there are a ton of cloud integration and other built-in technologies you can use later if you want to explore! Spring Boot is **POWERFUL**!

Click Next. Click Finish. Spring is now reaching out to an amazing service called Spring Initializr (with an odd spelling – no e!) to build our project skeleton for us. Unbelievable! It used to take weeks to get to this point in previous versions of this course and others!

It may need upwards of 10 minutes to download everything this first time, but I promise it will get much faster the more we use it. Maven'ed in repositories like the ones we selected in Spring Initializr are cached and stored locally for ultra-fast future compilations and builds. Please let it finish uninterrupted or you risk having to start again.

Expand the A1YourName project in the project explorer.

In the src/main/java folder is a class called A1YourNameApplication.java with a main method that "runs" our Spring Application. Additionally, in the src/test/java folder is a class called A1YourNameApplicationTests.java which will automatically run any JUnit tests we put in it. Junit is a testing technology largely covered in a previous course you had, and is very worth exploring so you can put it to use here too! Finally, in the src/main/resources folder is a file called application.properties which we'll use to configure our application, as well as storage for our static and template resources. Awesome!

Type

```
System.out.println("Hello World from <Your Name>!");
```

at the bottom of the A1YourNameApplication main method.

Type

```
Assertions.assertTrue(true);
```

inside the A1YourNameApplicationTests contextLoads(…) method.

Press Ctrl-Shift-O to Organize your Imports if it doesn't find the assertTrue method. Alternatively, you can hover over any code that is underlined in red and ask Eclipse to try to fix it for you. It definitely will not code everything for you automatically (obviously), but is super useful for quick fixes here and there **IF** (and **ONLY** if!) you know what you're doing. We want org.junit.jupiter.api.Assertions here – not any of the others.

Run the project as a Spring Boot Application by right-clicking on the project name and selecting Run As → Spring Boot Application.

Take a screenshot, save it as a JPEG or PNG with your name as the filename (**YourName.PNG**, for example), and upload it to the assignment submission folder. Saying it again – PLEASE USE **your NAME** as the **file name** and upload the JPEG or PNG directly! **Do not upload screenshots in a Word doc or a zip** of them or anything else. With images I can easily and directly connect to you through your filename, the marking on my end is greatly simplified. I will eventually deduct marks if you refuse to follow this simple procedure.

Run the test by right-clicking on the project name and selecting Run As → JUnit Test.

Click on the **JUnit** tab near the bottom, and make sure it's all green for good! Take a second screenshot, save it as a JPEG or PNG file with your name and a hyphen 2 (**YourName-2.PNG** or whatever) and upload it as part of your submission as well.

Again, please upload **actual images**, as opposed to Word Docs or zips or RARs or whatever with images inside them, and remember, **use your name as the image filename**!!!

That's it! Nicely done! 😊