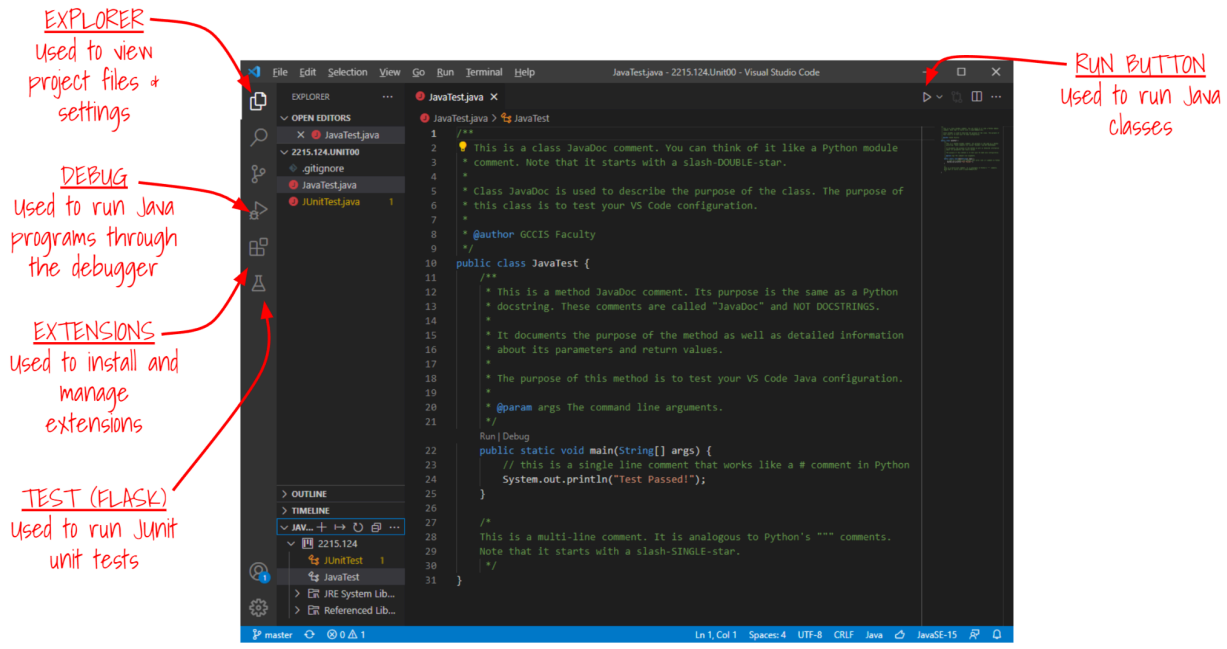
Software Dev. & Problem Solving II — GCIS-120/124

Pre-Course Setup — Assignment 0.0

Goals of the Assignment

The goal of this assignment is to install Java and configure VS Code for Java, Maven, and JUnit, the unit testing framework that you will be using this semester to write tests for your code. Taking the time to complete this assignment before the first lecture will allow you to get past any technical difficulties during the installation and configuration process so that you are ready to go on the first day of class. If you do not complete this assignment ***before*** your first lecture, you will not be able to complete the in-class activities or start assignment 1.1 right away. Please read the assignment ***in its entirety*** before asking for help.

It may have been a while since you last used VS Code to write a program. Here is a review of the layout of VS Code. Please refer to this image when completing the rest of this assignment.



The layout of the VS Code integrated development environment.

Video Guides

You may find the following video guides helpful when going through the activities listed below. These videos will show you how to configure your path for Java and Maven. ***Note***: these videos were recorded in 2023, so the version numbers in the video will not match what you see online. *Download the latest versions.*

1. Windows Tutorial
2. MacOS Tutorial - note that if you are using bash shell on your Mac you will need to edit the file named `.bash_profile` instead of `.zshenv`. If you are not sure which terminal program you are using, run the `ps` command from the terminal. It should indicate either `zsh` (`.zshenv`) or `bash` (`.bash_profile`). See the output examples below.

```
  PID TTY           TIME CMD
 1006 ttys000    0:00.17 -bash
 1042 ttys001    0:00.24 -bash
```

```
  PID TTY           TIME CMD
31879 ttys000    0:00.04 zsh
```

Activities

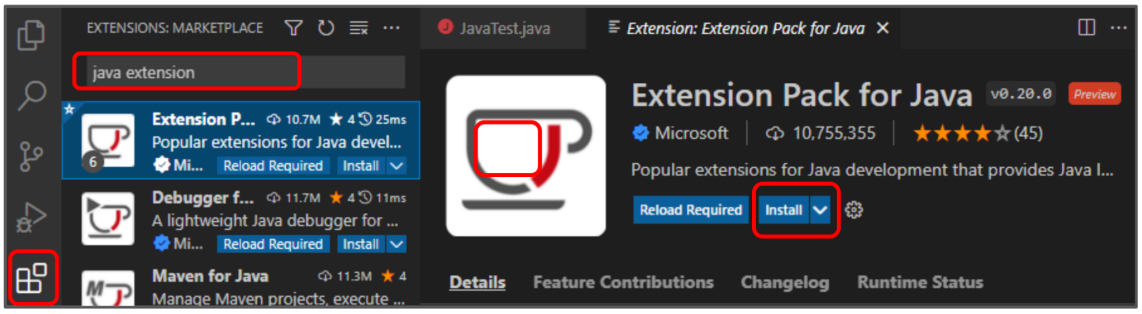
1. If you have not already done so, join the course Discord server. You will find the link to the server on MyCourses under *Content/Student Resources*. Once you have joined the server, post a message in the `#introduce-yourself` channel with your real first and last name and the name of your instructor.

2. Make sure that VS Code has been updated to the latest version using *Check for updates…* under the *Help* menu. Once you have verified that VS Code has been updated, close it before moving on to the next step.

3. Install the latest version of the ***Java SE Development Kit*** (***JDK***). You ***do not*** want to install the JRE (Java Runtime Environment) by itself as you will not be able to compile your programs!
    a. As of this writing, the latest version is Java 21. You can find it on the Java downloads page.
    b. Download the *installer* for your operating system, e.g. Windows or macOS.
    c. Run the installer and accept all of the default options in the installer.
    d. Verify that you have installed the correct version of Java by launching the terminal and running:
        i. `java -version`
        ii. `javac -version`

        You should see the same version number for both. If you see the wrong version for one or both, or one of the commands is not recognized, you will need to update your system path so that it includes your new Java install location.

4. Once you have installed Java and verified that it is working from the command line, you will want to make sure that the ***Extension Pack for Java*** is installed in VS Code.
    a. Launch VS Code.
    b. Click on ***Extensions*** in the sidebar. If you are having trouble finding it, refer to the layout diagram above.
    c. Type "Java Extension Pack" into the search field at the top and click on it.
    d. If it is already installed, you are all set! If not, install it and restart VS Code.



5. Next, you will need to install Maven, a build tool used to compile and run Java projects. Installing Maven is a bit of a manual process that requires you to download and extract a ZIP file and then update your system path to include the installation location.
    a. Open the Maven downloads page and download the **Binary zip archive**. (download the file in the **Link** column).
    b. Extract the archive into an appropriate location for your operating system, e.g. the `/c/Program Files/` directory on Windows or in the `Applications` folder on macOS.
    c. In order to run Maven from the command line, you will next need to add the Maven `bin` directory to your system path (this is where the `mvn` executable is located). You can find it inside the folder that you downloaded and extracted, e.g. `C:\Program Files\apache-maven-3.9.6\bin`. Make sure that you know the exact path to the Maven bin folder and add it to your system path.
    d. Maven needs to know where Java is located on your operating system. You will need to create an environment variable named `JAVA_HOME` that refers to the Java installation directory, e.g. `C:\Program Files\Java\jdk-21` on Windows.
    e. Test your configuration by running `mvn -version` from the command line. You should see output like the example below. The Java version and path should match your Java installation.

```
user@computer MINGW64 ~
$ mvn -version
Apache Maven 3.8.6 (84538c9988a25aec085021c365c560670ad80f63)
Maven home: C:\Program Files\apache-maven-3.8.6
Java version: 19.0.1, vendor: Oracle Corporation, runtime:
C:\Program Files\Java\jdk-19
Default locale: en_US, platform encoding: UTF-8
OS name: "windows 11", version: "10.0", arch: "amd64", family:
"windows"
```

6. Accept the GitHub Classroom assignment.
    a. Once your repository has been created, use Git to clone the repository to your computer.
    b. Launch VS Code and open the repository.
    c. Open a Git Bash terminal and execute the following commands:
        i. `mvn clean compile exec:java`
        ii. `mvn test`

   d. You should see output after running both commands indicating **`BUILD SUCCESS`**. This means that your environment is working properly!
   e. If you do not see that the build was successful, post your errors and/or questions on the course Discord server. Either way, take a screenshot of the output in the terminal and save it to your repository.

7. Create a text file named "`about_me.txt`" and write your full name and hometown in the file. Use Git to push it to your repository.