

This course uses Java as the language of instruction. The following instructions will guide you through installing the Java programming environment.

# 1 General Setup

The first part of setting up the environment is the same regardless of which operating system you use.

1. First, install a Java Development Kit (JDK)

- Go to <https://adoptium.net/> and click the download button labeled “Latest LTS Release”.
- After it’s done downloading, run the installer. Accept the default settings.

2. Next, Install the course libraries.

- Go to <https://kyleclapper.com/cs210/> and download the course folder.
- After it’s done downloading, unzip the folder and put it in your in your home directory. This directory is the same name as your MacOS or Windows username. On MacOS, the path to your home folder should be like `/Users/<your username>`. After moving the course folder, the path to it should be `/Users/<your username>/cs210`. On Windows, the path to your home folder should be like `C:\Users\<your username>`. After moving the course folder, the path to it should be `C:\Users\<your username>\cs210`.

3. Next, install IntelliJ IDEA Community Edition. This is the integrated development environment (IDE) we’ll be using to write Java programs.

- Go to <https://www.jetbrains.com/idea/download/>. Scroll down to the **Community Edition** section and click download. *Do NOT download the Ultimate Edition.*
- After it’s done downloading, run the installer. Accept the default settings.

4. Next, configure IntelliJ to use the JDK we downloaded.

- Open IntelliJ. Click “Open” then select the “dummy\_project” folder. You can find it in the course folder under `cs210/workspace/dummy_project`.
- Go to File >> Project Structure.
- Click on the “Project” section on the left.
- Click on the “SDK” dropdown menu and select the JDK we installed.
- Click “Apply” and then “OK”.

5. Finally, add the course libraries to IntelliJ.

- With IntelliJ still open to the “dummy\_project” workspace, click on File >> Project Structure.
- Click on the “Global Libraries” section on the left.
- Click on the “+” icon at the top left and select “Java”.
- Browse to the course folder, then select **BOTH** the “dsa.jar” and “stdlib.jar” files from the “lib” folder. To select both, click one, then hold the control key (or command on MacOS) and click the other.
- Click “Open”/”OK” then “OK” then “Apply” then “OK”.

## 2 Environment Variable Setup

Environment variables are variables in your terminal that can be read by programs like the Java compiler. We need to configure two environment variables, `PATH` and `CLASSPATH`.

The `PATH` environment variable is a list of directories where programs are found. When you try to run a program on the command line, it'll look in these directories for the program with the same name you entered. The class folder comes with a program called “check\_style”. In order to use it properly, we need to add the “bin” folder to the existing `PATH` variable.

The `CLASSPATH` variable is used by Java to determine where to find compiled Java programs. We need to create this variable.

The process of configuring these environment variables is different depending on whether you're using Windows, MacOS, or Linux.

### Windows

Open Powershell or a terminal in IntelliJ (File >> View >> Tool Windows >> Terminal). Then enter the following commands. Do not enter any line that starts with the `#` symbol. These are comments and only serve to explain what you're doing. Do not enter the `$` at the start of the lines.

```
>_ ~/cs210
# Change the directory to the class folder.
$ cd $HOME\cs210

# Set the CLASSPATH environment variable.
$ setx CLASSPATH ".;\out;$HOME\cs210\lib\stdlib.jar;$HOME\cs210\lib\dsa.jar"

# Update the PATH environment variable.
$ setx PATH "$env:PATH;$HOME\cs210\bin"
```

### MacOS

Open Terminal or a terminal in IntelliJ (File >> View >> Tool Windows >> Terminal). Then enter the following commands. Do not enter any line that starts with the `#` symbol. These are comments and only serve to explain what you're doing. Do not enter the `$` at the start of the lines.

```
>_ ~/cs210
# Change directory to the class folder.
$ cd $HOME/cs210

# Make the check_style program executable.
$ chmod 755 bin/check_style

# Set the CLASSPATH environment variable.
$ echo "export CLASSPATH=./out:$HOME/cs210/lib/stdlib.jar:$HOME/cs210/lib/dsa.jar" >> $HOME/.zshrc

# Update the PATH environment variable.
$ echo "export PATH=$HOME/cs210/bin:$PATH" >> $HOME/.zshrc
```

### Linux

Open Terminal or a terminal in IntelliJ (File >> View >> Tool Windows >> Terminal). Then enter the following commands. Do not enter any line that starts with the # symbol. These are comments and only serve to explain what you're doing. Do not enter the \$ at the start of the lines.

```
>_ ~/cs210
# Change directory to the class folder.
$ cd $HOME/cs210

# Make the check_style program executable.
$ chmod 755 bin/check_style

# Set the CLASSPATH environment variable.
$ echo "export CLASSPATH=../out:$HOME/cs210/lib/stdlib.jar:$HOME/cs210/lib/dsa.jar" >> $HOME/.bash_profile

# Update the PATH environment variable.
$ echo "export PATH=$HOME/cs210/bin:$PATH" >> $HOME/.bash_profile
```

### 3 Testing Your Setup

Test your programming environment to make sure it's working correctly. First, restart your terminal. If you're using the terminal within IntelliJ, restart IntelliJ instead. Open a terminal and navigate to the "dummy\_project" folder. Then enter the following commands (do not enter the \$ symbol):

```
>_ ~/cs210/workspace/dummy_project
$ javac -d out src/Main.java
$ java Main
```

If the program printed the words "Hello World" to the screen, then you've successfully set up your programming environment. If it doesn't, please ask an instructor for help.