

# Lab 0: Hello World

CSCI 136: Spring 2024

---

## Learning objectives

---

1. **Software Development:** Set-up your development environment. Use the command-line and git.
2. **Programming:** Practice compiling and running a simple Java program.
3. **Programming:** Practice basic Java syntax, including if-statements and for-loops.

---

## Classroom management tools

---

In CS 136, we will use the following classroom management tools. If you have difficulty logging in, please contact the course staff.

### 1 Piazza

Piazza is a classroom management tool that allows for quick and efficient answering of questions as well as the ability to post anonymously.

You should have received an email that allows you to access Piazza via this link: <https://piazza.com/class/lrt7ezh09k71j5/>. Navigate to that link and ensure that you are able to log in.

### 2 GitHub Classroom

We will use GitHub Classroom to distribute the starter code for the lab assignments. Later, we will walk you through setting up a GitHub account and accessing the starter code for this lab assignment.

### 3 Gradescope

Gradescope is an online grading software that allows students to receive faster and more detailed feedback on their work, and allows instructors to see detailed assignment and question analytics.

When you (the student) submit an assignment, Gradescope spins a virtual (Docker) container to compile and execute your code. It then runs a suite of tests to autograde portions of your assignments.

You should have received an email enrolling you in Gradescope. If you did not receive the email or have difficulty logging in, please contact the course staff.

---

## Development environment

---

We will be using and supporting the following development environment for CS 136:

1. Java 8 (Amazon Corretto 8)
2. Sublime (text editor)
3. The Terminal (iTerm2; zsh)

## Log-in to the machines in TCL 217

- Name: Your UNIX username (e.g., Katie's is kak5 since her email is kak5@williams.edu)
- Password: Same password you use for your Williams email

*Note:* On the lab machines, there is no access to a networked home directory. Each machine has just a local home directory, and users cannot rely on the local directory to be retained. Please use the steps to add, commit and push files via Git to maintain a record of your work.

## Optional: Development environment on your own machine

This step is optional (but recommended) to download software on your local machine. Of course, you are welcome and able to use the lab machines for the entirety of the semester.

### 4 XCode

Make sure you have XCode installed on your machine. Navigate to the AppStore and search “XCode.” If you do not have it installed, install it.

### 5 The Terminal

On macOS, the “Terminal” is a text-based interface that allows users to interact with the computer’s operating system or other programs by typing commands and receiving text output in return. The “command line” refers to the actual line in a terminal or console window where you type your commands. In this class on Macs, we will use iTerm2 and **zsh** (pronounced Z-shell).

To install iTerm2 on your Mac, follow [this link](#) to download.

Once you have iTerm2 downloaded and installed, open iTerm2 by pressing Command+Space (to open a Spotlight search) and then type iTerm2.

To check that you are using zsh, type the following into the terminal

```
ps -p $$
```

You should see “-zsh” under “CMD”.

We assume basic knowledge of command-line skills. If this is something you need more practice with, we recommend browsing <https://missing.csail.mit.edu/>.

### 6 Java

We will install Java 8 on your machine.

1. Uninstall any versions of Java already on your machine.

- (a) Open a Terminal and run

```
java -version
```

- (b) If you get an error, this means you don’t have Java and you can continue to the next step.
  - (c) Otherwise, we need to uninstall Java. On a Mac, open a terminal and copy and paste the following commands

```
sudo rm -rf /Library/Internet\ Plug-Ins/JavaAppletPlugin.plugin
sudo rm -rf /Library/PreferencePanes/JavaControlPanel.prefPane
sudo rm -rf /Library/Application\ Support/Oracle/Java/
sudo rm -rf /Library/Java/JavaVirtualMachines
```

2. Download and Install the Amazon Corretto 8 JDK

- (a) **Mac:** First click the Apple icon at the top left of your screen and select ‘About This Mac’
  - i. M1 or M2 Chip: Use this download <https://corretto.aws/downloads/latest/amazon-corretto-8-aarch64-macos-jdk.pkg>
  - ii. Intel Chip: Use this download <https://corretto.aws/downloads/latest/amazon-corretto-8-x64-macos-jdk.pkg>
- (b) To run the installer, click the file that you just downloaded (on Mac, it will be in your Downloads folder) and then click all the way through the Installer.

### 3. Check whether you were successful by opening a Terminal and running

```
java -version
```

- (a) On Katie’s Mac here is the output (although it is ok if yours is slightly different)

```
openjdk version "1.8.0_392"  
OpenJDK Runtime Environment Corretto-8.392.08.1 (build 1.8.0_392-b08)  
OpenJDK 64-Bit Server VM Corretto-8.392.08.1 (build 25.392-b08, mixed mode)
```

- (b) If you don’t see version 1.8 or “Corretto” please ask one of the instructors or TAs for help.

## 7 Sublime

1. Download Sublime for your operating system (Mac, Linux or Windows) via this link <https://www.sublimetext.com/>
2. Drag Sublime from your Downloads folder to your Applications folder.
3. We will add a shortcut to your Terminal profile to make it easier to open files via the command-line. Follow [these instructions](#).
4. Then quit your Terminal and open it again (to reload your profile file).

Then you should be able to open a Sublime editor directly from the Terminal. For example to open a folder, type

```
subl -n .
```

## Cloning & editing the lab repository

In this class, we will use GitHub Classroom to distribute starter code. Complete the following steps to access the starter code for this lab.

## 8 Create a GitHub account

*Notes:*

- If you already have a GitHub account, please use your existing one.
- This is a Github account and not a GitLab account (what you might have used if you were in CS 134 here at Williams).

To set-up a GitHub account

1. Navigate to <https://github.com/> and click “Sign Up” in the upper right-hand corner of the page.
2. Enter your Williams email address and click continue. Create a password and click continue.
3. Create a username and click continue. Follow other steps to verify your account.
4. Open your Williams email to get the verification code. Enter this to verify your account.
5. At the very last step, click “Continue for Free”.

## 9 Set-up GitHub SSH keys

We will set-up SSH keys to allow you to push and pull quickly and easily to GitHub.

1. Navigate to [this page](#) and follow the instructions to set-up an SSH key on your local machine.
2. Then navigate to [this page](#) to add the SSH key to **authenticate** your GitHub account (you do not need to add a key for “commit signing” unless you really want to).

## 10 Open repository

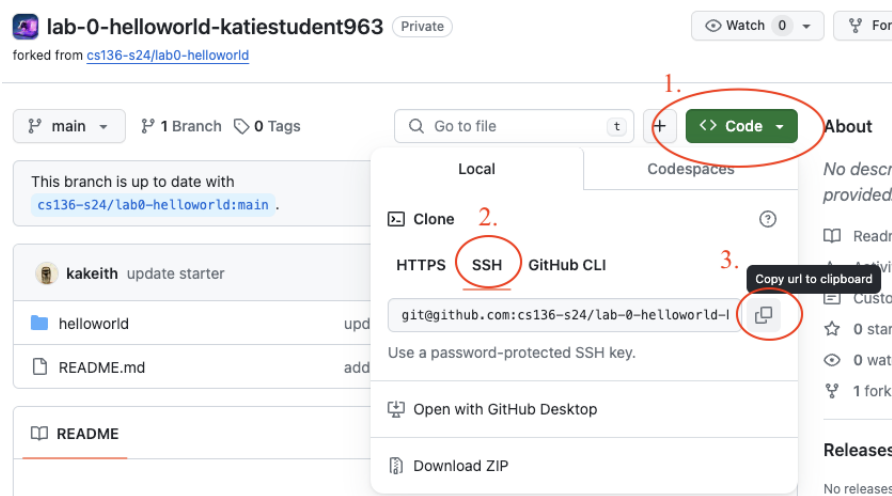
The course staff have sent you an invite link in an email/Piazza post with the subject line “CS 136 Lab 0 Invite Link”.

Copy and paste the link that was sent. A screen should pop up that says “GitHub Classroom by GitHub would like permission to:”. When you see this screen, follow these steps:

1. Click the green “Authorize GitHub Classroom” button.
2. Select your email as the “Identifier.”
3. Click “Accept this assignment”
4. Wait 1-2 minutes for your repository to be configured. After 1-2 minutes click refresh if the page does not do so automatically.
5. When it is completed, you will see a page “You’re ready to go!”
  - (a) If you do not see this page, navigate to `github.com`. As long as you are logged in, you should see your new cloned repository on the left-hand side of the window.
6. Click the link to your repository on the page. For example, it will be something similar to <https://github.com/cs136-s24/lab-0-helloworld-katie> where `katie` is whatever your username is.

## 11 Clone the repo locally

Follow the three steps in the image below to copy the SSH link to your clipboard



(Optional: On your own laptop) Make a directory where you will store all your labs. For example,

```
cd ~
mkdir cs136
cd cs136
```

Navigate to the place you want to put the lab starter code and clone the GitHub classroom repository

```
git clone LINK-YOU-COPIED
```

replacing **LINK-YOU-COPIED** with the one you copied from GitHub (via the image).

Now move to that repository repository

```
cd lab-0-helloworld-YOURUSERNAME
```

where you replace **YOURUSERNAME** with your username on Github.

## 12 Editing

First check that you are in the correct repository by typing

```
pwd
```

On Katie's local machine this looks like

```
/Users/katherinekeith/lab-0-helloworld-katiestudent963
```

Then open the file with Sublime

```
subl ./helloworld/Hello.java
```

Read through the README.md for the instructions about the lab (this will render more nicely if you look at the version on Github). Start editing your file!

## 13 Version control: Git

When you have made sufficient changes, make sure to commit those changes to GitHub using git. For example,

```
cd helloworld/  
git add Hello.java  
git commit -m 'CHANGE TO YOUR MESSAGE'  
git push
```

If Git is new to you and you'd like more explanation and practice, we recommend browsing <https://missing.csail.mit.edu/>