Programming Assignment #0

Due: 9/7/25 @ 11:59pm

Assignment Link: https://classroom.github.com/a/veoTX_7d

Objectives

1. Setup your GitHub account (if necessary)
2. Setup IntelliJ to work with Java and your GitHub account
3. Confirm you can execute provided code and tests
4. Confirm that Autolab is working for you
5. Confirm that you are able to submit assignments via Git

Useful Links

1. Sign up for GitHub
2. Install IntelliJ
3. Testing with JUnit
4. Setup GitHub authentication
5. Autolab
6. Git documentation

Late Policy and Grading

Late submissions will not be accepted for this lab.
This lab is worth 0 points, but you must complete this assignment to get a passing grade.

Submissions close at 11:59pm on 9/7/25 and no late submissions will be accepted.

Instructions

In this lab you will complete the following steps to ensure that your development environment is set up appropriately, and you have linked your GitHub and UBIT together in Autolab. Each step is described in more detail in the sections that follow.

1. Ensure that you have an existing GitHub account, or create a new one if you do not
2. Ensure that you have an SSH public/private keypair and that it is registered with your GitHub account
3. Accept the PA0 assignment on GitHub Classroom by clicking the link above
4. Install Java and IntelliJ
5. Create an IntelliJ project from your PA0 repository
6. Edit `Main.java` to contain your UBIT name and your GitHub username
7. Edit `MainTest.java` to get a basic understanding of tests
8. Commit your changes and push them to GitHub
9. Submit to Autolab with your GitHub account

1 GitHub Account

If you do not have a GitHub account, sign up for one [here](#)

2 GitHub Authentication

To make sure you can checkout, commit, and push to your GitHub repository for each lab, you must set up private key authentication on the machine you will be using for development. The best way to do this is to follow the instructions on GitHub [here](#). Note that there are different versions of the instructions for Mac, Windows, and Linux systems.

After generating new SSH keys, you will have two files, a public key (ends in .pub) and a private key (no file extension). You must take the contents of the **public key** file and upload them to GitHub under your account settings.

3 Accept the PA0 Assignment

By clicking the link at the top of this writeup (https://classroom.github.com/a/veoTX_7d) you will have your own personal git repository created with the initial programming assignment code. The initial creation may take a minute or so, but once the repository is created you can view it on GitHub at the specified link.

4 Install IntelliJ

To make sure you can edit, compile, and run the various programming assignments, we suggest you use the IntelliJ IDE. In order to do so, you must have a version of the Java SDK (an up to date version can be downloaded [here](#)). IntelliJ can be downloaded [here](#).

5 Clone your PA repository into an IntelliJ Project

For this, and all subsequent programming assignments, you will need to clone the repository you created in step 3 into IntelliJ so you can edit, compile, run, test, and commit changes.

To start, create a new project in IntelliJ **from VCS** (VCS here stands for Version Control System)
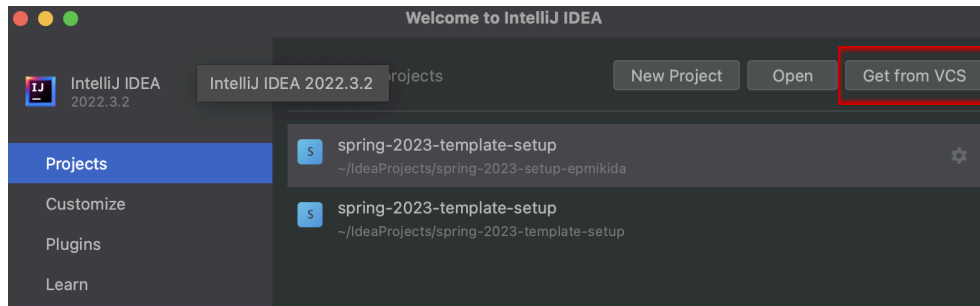
The subsequent dialog will ask for the URL of your repository. To find this, go to the GitHub page for your repository, and click the green code button, then copy the SSH URL.
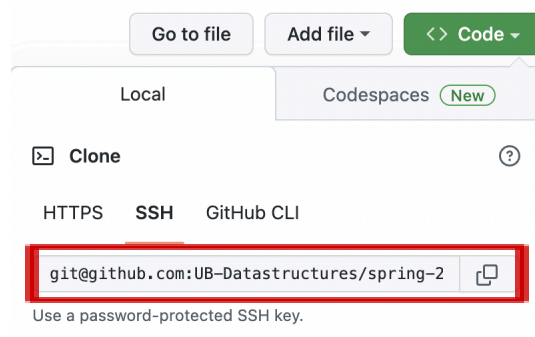
If the process works and the project is created successfully then you are ready to edit code! If not, you likely have an authentication error and need to go back to step 2.

6 Edit and Run Main.java

Once you've created the project, open up `Main.java` (located in `src/cse250.pa0` from the file explorer on the left side of the window). In this file, there are two specific variables we want you to change. The variables `YOUR_UBIT` and `YOUR_GITHUB` (on lines 5 and 6) are strings that are initially assigned the empty string.

Modify the string assigned to `YOUR_UBIT` to be your UBIT name.
Modify the string assigned to `YOUR_GITHUB` to be your GitHub username.

For example, my lines 5 and 6 would be the following:

```
static String YOUR_UBIT   = "epmikida";
static String YOUR_GITHUB = "epmikida";
```

Once you've finished editing the file you can run it in one of two ways to verify that everything worked correctly. The first option is to click the green play button in the left margin of either the `Main` class, or the `main` method.

The second method is to run the file by clicking the green play button in the top right of the IntelliJ window. Make sure that you have "Main.java" or "Current File" selected. The code should run without error and you should see your UBIT and GitHub usernames printed in the terminal.

7 Edit MainTest.java

After editing `Main.java` it's time to edit `MainTest.java` (located in `src/cse250.pa0/tests`). This file contains a single `JUnit` test that is meant to test that `Main` is working as expected, but there are some issues with the test that you need to fix.

Before changing anything, run the tests to see what failed tests look like. You can run a single test by clicking the play button in the margin of the test function, or run all of the tests in the file by clicking the play button in the margin of the class or in the top right of the IntelliJ window.

Once you've run the tests and have seen them fail, update the assertions to work properly, and re-run the tests to verify that they pass. They must pass when you submit to AutoLab to get credit for this assignment.

8 Commit and Push

Assuming you have successfully completed step 7, you are now ready to commit your changes and push them to your repository. This is equivalent to saving your changes and updating the remotely hosted version of your repository to contain the changes you just made on your local machine. This allows others (like Autolab) to see the most up-to-date version of your code.

Along the left side of the IntelliJ window, open the Commit tab.

Use the checkboxes, select the changes that you want to commit (in this case it should be everything).

Enter a MEANINGFUL commit message, and click the commit and push button. Your code has now been pushed, and if you go back to your GitHub repository in your browser you should be able to see the changes.
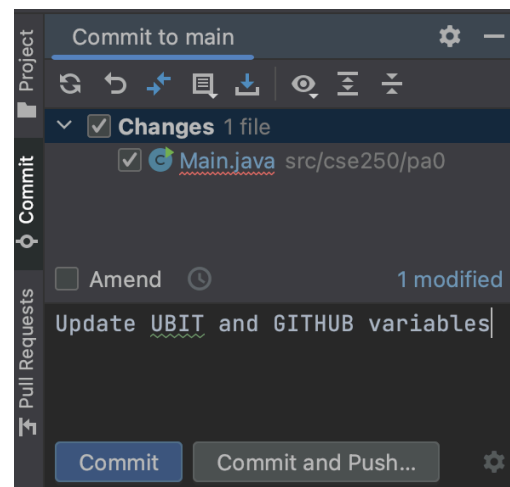
Some useful advice:
As projects get more complicated, you will end up making many changes to your code. Some

will work, some will not. Get in the habit of committing your changes often, and using descriptive messages.

It is not uncommon (no matter your experience level) that you will make a number of changes, and then find out they have broken your code in ways you did not expect. Many times, these changes will not be straightforward to undo (you likely won't remember every last detail of your changes).

If you have been committing your code at regular intervals, and used good commit messages, you can revert your code to a previous commit trivially. This can save you a significant amount of time, and even points on your grade (for example if you had a partially working code, and have made changes that have caused your code to crash and score 0 points).

9 Submit to Autolab

Finally, you are ready to submit to Autolab to verify that everything works. In your browser, navigate to the class Autolab ([here](#)), and click on the PA0 assignment.

Every time you submit to any assignment, you must check the box to verify that you understand the course policy on Academic Integrity. This policy is detailed in the course syllabus. Once you have verified you understand the Academic Integrity policy, select the GITHUB submission option. If this is your first time submitting to Autolab via GitHub, you will first have to connect your GitHub account. Once your GitHub account has been connected you will not have to connect it again in the future.

Once your GitHub account is connected to Autolab, select the repository containing your PA0 assignment, select the main branch, and select the most recent commit, and click submit.

After submitting, Autolab will automatically grade your assignment. This should only take 30 seconds or so, at which point you can refresh and see your grade. You must get a 1.0/1.0 on this assignment to have any future coursework graded. If you do not get a 1.0, you can click on your grade to see feedback about what went wrong, make adjustments, and resubmit.

🕐 Due: September 10th 2023, 11:59 pm EDT (UTC -04:00)	
📅 Last day to handin: September 10th 2023, 11:59 pm EDT (UTC -04:00)	

UPLOAD	GITHUB

Repository

UB-Datastructures/fall2023-pa0-setup-epmikida

Branch

main

Commit

ea359a8 (Add UBIT and GitHub)

✓ I affirm that I have complied with this course's academic integrity policy as defined in the syllabus.

SUBMIT

(unlimited submissions left)