CS 61B

Resources / Guides / Assignment Workflow

Assignment Workflow

Author: Ethan Ordentlich

Assignment Workflow

This guide describes how to set up an assignment in CS 61B.

Getting the Skeleton

A video demo of this guide is available at this link.

The `skeleton` remote repository contains the skeleton code for all assignments. Whenever a new assignment is released, or if we need to update an assignment, you will pull from the skeleton. First ensure that you are in your `sp24-s***` repository directory, then:

```
git pull skeleton main
```

Copy

This fetches all remote files from the repo named `skeleton` (which is located at `https://github.com/Berkeley-CS61B/skeleton-sp24.git`) and copies them into your current folder.

> **WARNING**
>
> If you get an error similar to `fatal: refusing to merge unrelated histories` , you can fix this each time by using
>
> ```
> git pull --no-rebase --allow-unrelated-histories skeleton main
> ```
>
> Copy

(If you’re working on Lab 1, go back to the lab spec at this point.)

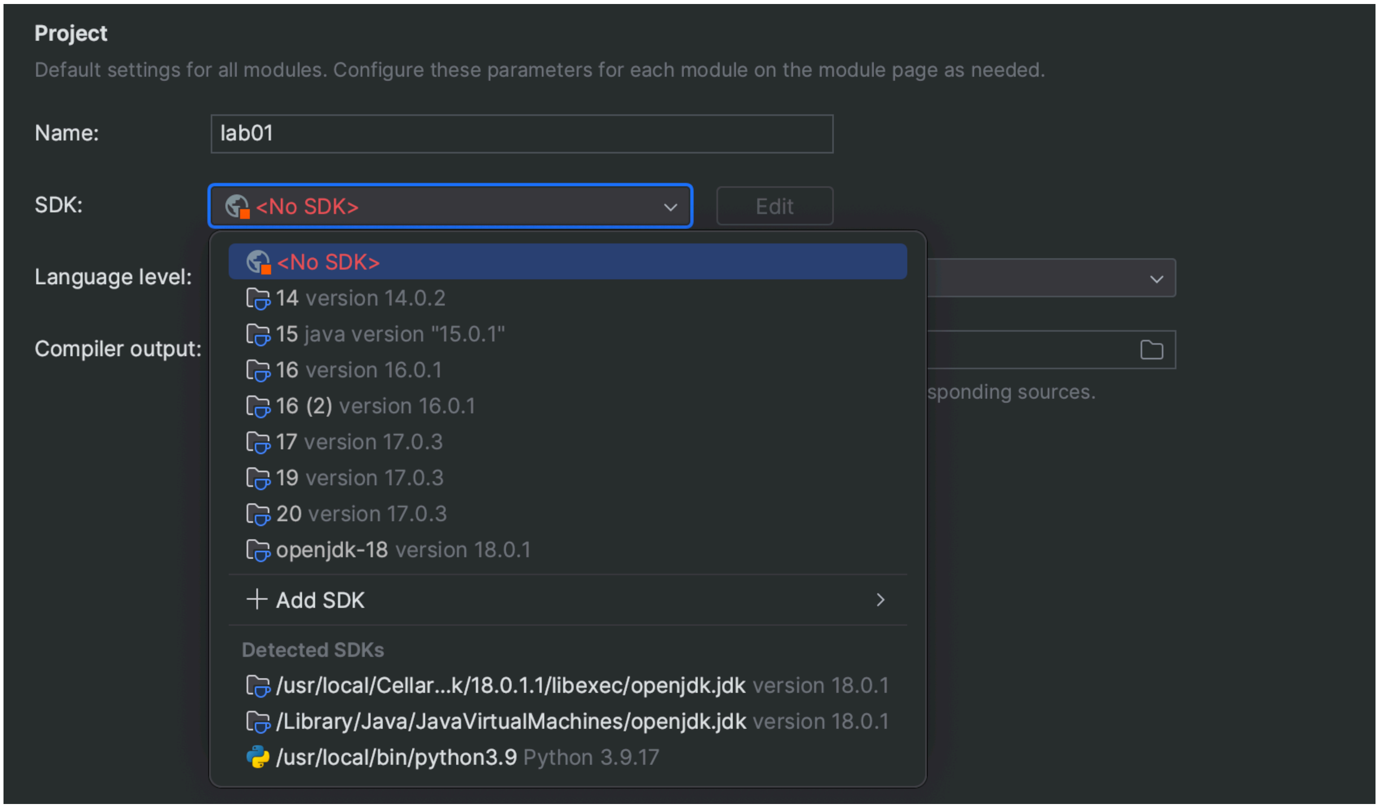
Opening in IntelliJ

The following instructions apply for **every** assignment. **Each time after pulling from** `skeleton` **to get new lab or project files, you will need to run through the following steps again.**

1. Start up IntelliJ. **If you have no projects open**, click the "Open" button. If you have a project currently open, navigate to **"File –> Open"**.

2. Find and choose the directory of your current assignment. For example, for Lab 1, you would select the `lab01` directory inside your `sp24-s***` repo.

3. Navigate to the **"File -> Project Structure"** menu, and make sure you are in the **Project** tab. Set your project SDK to your installed Java version. If 17 or higher isn't available in the dropdown, make sure you downloaded and installed Java completely.



Project

Default settings for all modules. Configure these parameters for each module on the module page as needed.

Name: lab01

SDK: <No SDK>

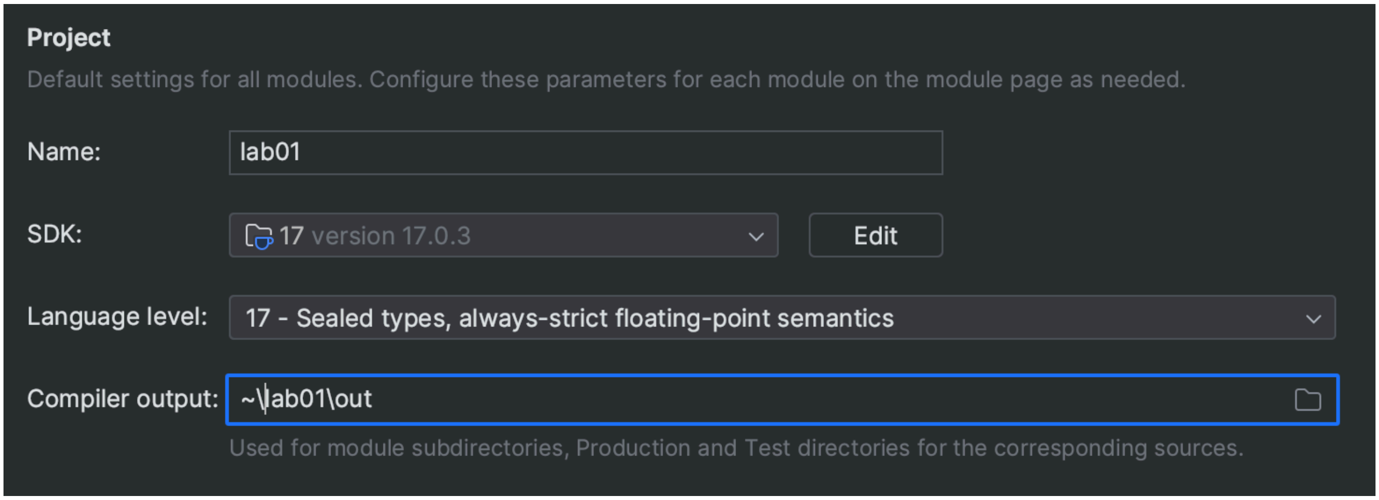
Language level:

Compiler output:

Edit
<No SDK>
14 version 14.0.2
15 java version "15.0.1"
16 version 16.0.1
16 (2) version 16.0.1
17 version 17.0.3
19 version 17.0.3
20 version 17.0.3
openjdk-18 version 18.0.1
Add SDK
Detected SDKs
/usr/local/Cellar...k/18.0.1.1/libexec/openjdk.jdk version 18.0.1
/Library/Java/JavaVirtualMachines/openjdk.jdk version 18.0.1
/usr/local/bin/python3.9 Python 3.9.17
sponding sources.

4. Still in the **Project** tab, set the Project Language Level to "17 - Sealed types, always-strict floating-point semantics".

   At this point, the Project tab should look something like this:



Project
Default settings for all modules. Configure these parameters for each module on the module page as needed.

Name: lab01

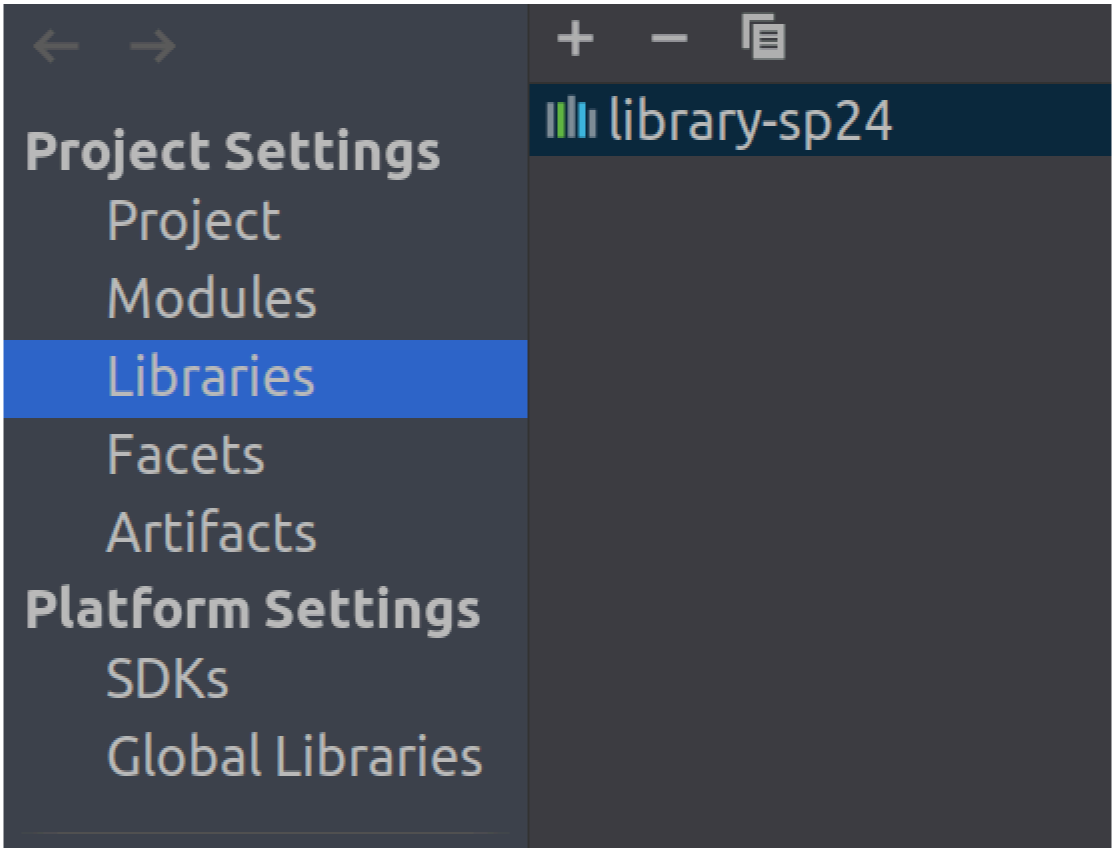
SDK: 17 version 17.0.3

Language level: 17 - Sealed types, always-strict floating-point semantics

Compiler output: ~\lab01\out
Used for module subdirectories, Production and Test directories for the corresponding sources.

- The SDK is set to Java 17 or higher
- The Language level is at least 17, and at most the SDK.
- The Compiler output is filled in, and is set to the assignment directory, followed by `out`

5 Still in Project Structure, go to the **Libraries** tab. Click the "**+ -> Java**" button, then navigate to `library-sp24`, select the folder, and click "Ok".



6 Click "Ok" to apply your settings and leave Project Structure.

At this point, if things have been configured correctly:

- Each Java file should have a blue circle next to its name.
- When you open any file, none of the code should be highlighted in red.

Submitting to Gradescope

1. Add your assignment directory using `git add`. For example, for Lab 1, from your repo root (`sp24-***`) you would use `git add lab01`. From the assignment directory, you could use `git add .`.
2. Commit the files using `git commit -m "<commit message here>"`. The commit message is required. For example, `git commit -m "Finished Lab 1"`.
3. Push your code to your remote repository with `git push origin main`.
4. Open the assignment on Gradescope. Select Github, then your `sp24-s***` repository and the `main` branch, then submit your assignment. You will receive a confirmation email, and the autograder will run automatically.

Gradescope will use the latest version of your code from Github. **If you think that Gradescope isn't grading the right code, check that you have added, committed, and pushed with** `git status`**.**

If you've gotten yourself into a situation where you're unable to push for some reason, see Git WTFS.