

CMPT 280

Tutorial: Obtaining and Setting Up lib280 (IntelliJ Version).

Mark G. Eramian

University of Saskatchewan

Before you Start

- This tutorial assumes that you have the IntelliJ IDE installed and working on your computer.
- If you do not, follow these steps first:
 - Install a Java Software Development Kit on your computer. This can be downloaded from Oracle (look for the "Java SE Development Kit 8" download).
 - Download the IntelliJ IDEA Community Edition from: <https://www.jetbrains.com/idea/>. If you have the Ultimate Edition that is fine, but the Community Edition will suffice for this course.
- Note: The screenshots shown in this tutorial are from IntelliJ 2021.3 Ultimate Edition.

IntelliJ IDEA Project Organization

- In IntelliJ you have a *project* that can be organized into *modules*.
- You can create modules for each assignment.
- You'll also create a module for a new version of lib280 for each assignment.

The Rest of the Tutorial

In the rest of this self-guided tutorial we will do the following:

1. Set up an empty project.
(You only need to do this once for the entire course.)
2. Learn how to import versions of lib280 as a module.
(You'll need to do this once for each assignment.)
3. Learn how to create a new module that can contain a program
that can access the code in a lib280 module.
(You'll need to do this once for each assignment question that
requires you write a program that is separate from lib280.)

Part I

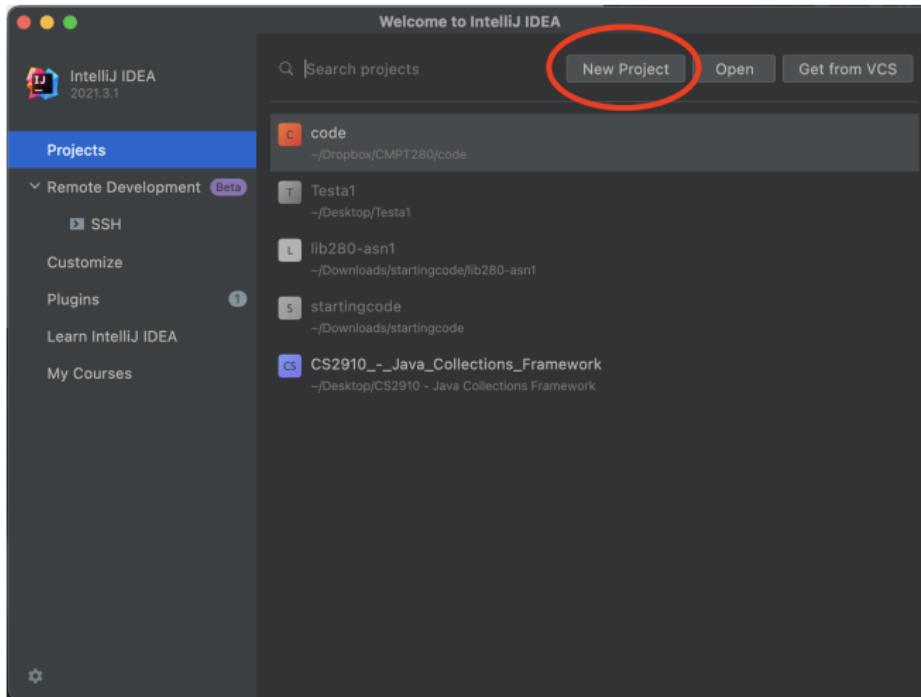
Setting up an Empty Project

Setting up and Empty Project

We will now set up an empty project in IntelliJ. **You will only have to do this once for the whole course.** All of your subsequent versions of lib280 and programs that you will write will be modules within this project.

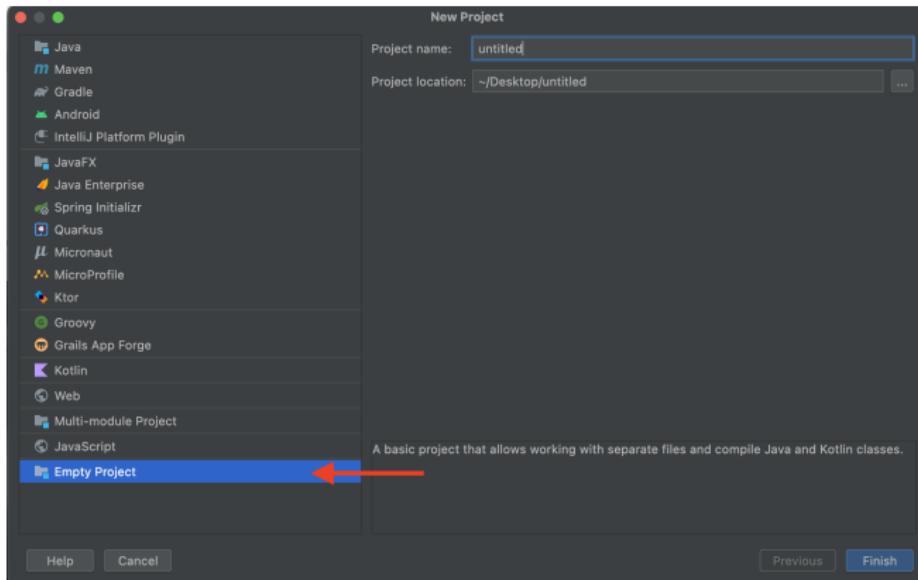
Step 1: Open IntelliJ and Create a Project

Open IntelliJ and choose "Create new Project".



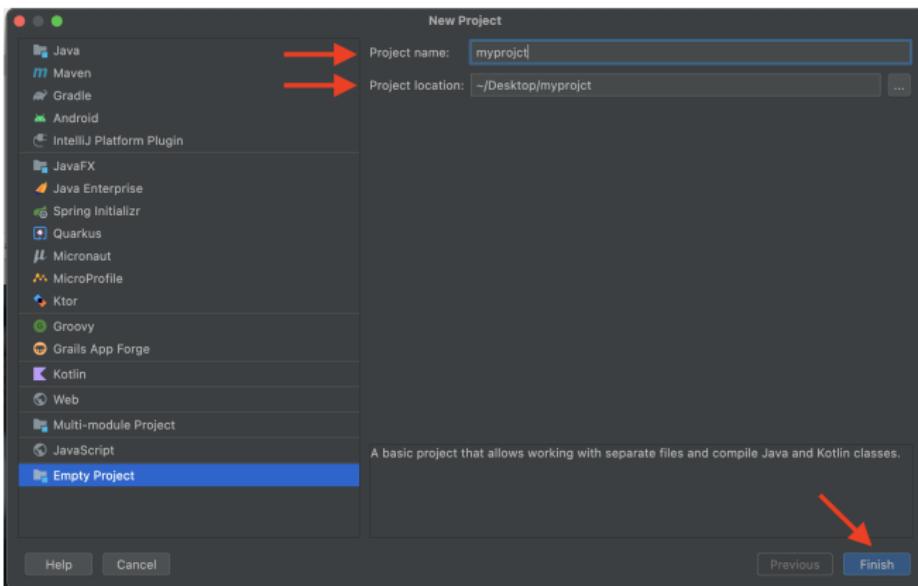
Step 2: Select Project Type

Scroll to the bottom of the list on the left and select "Empty Project".



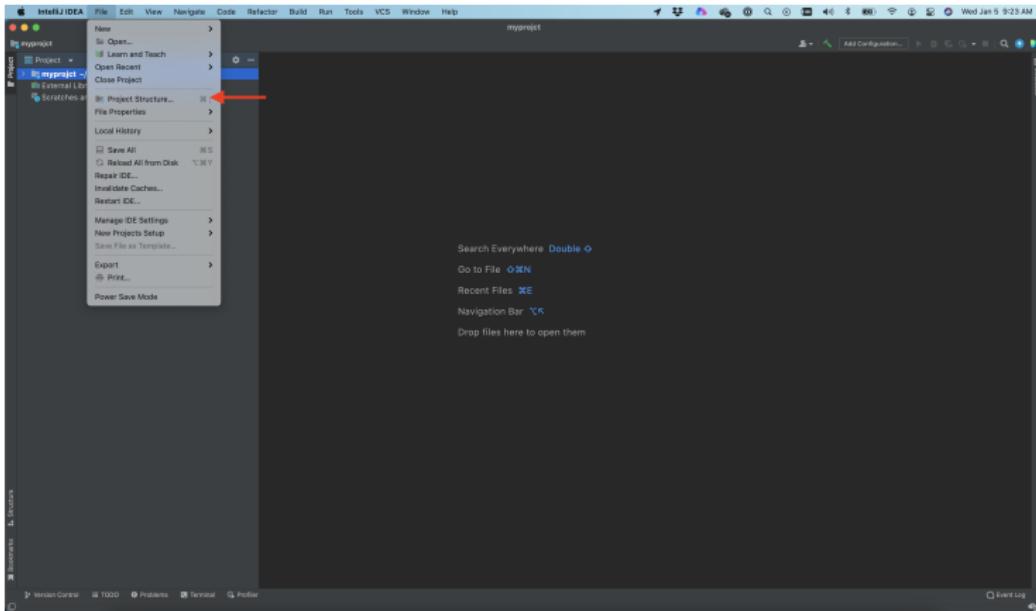
Step 3: Choose Project Location

Enter the project's name and enter the folder name where you'd like your project to reside in the places shown. Project location can be a folder that does not yet exist. Click "Finish". If IntelliJ asks if it's OK to create the folder you entered, click "OK".



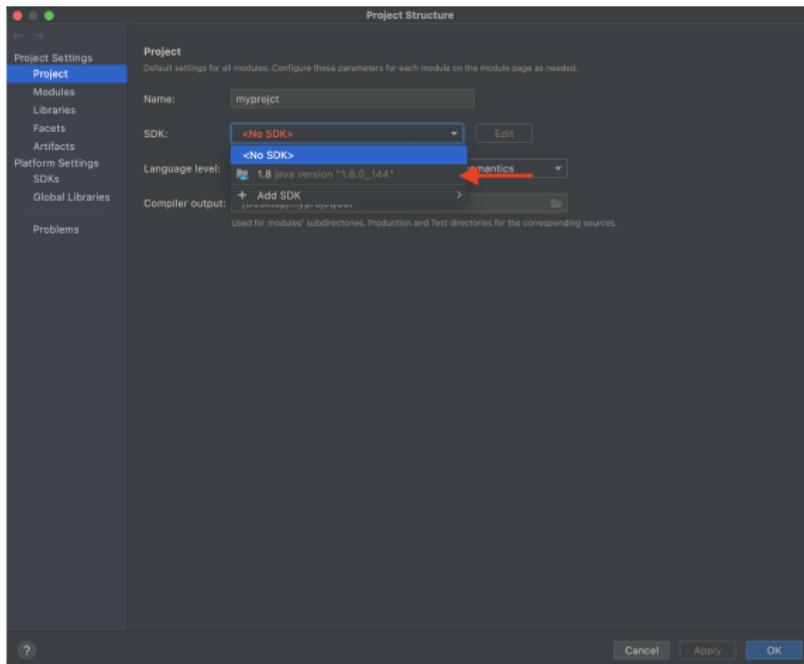
Step 4: Select the Java SDK

You should now see your empty project, but we need to specify which Java SDK to use. Go to the FILE menu and select “Project Structure”.



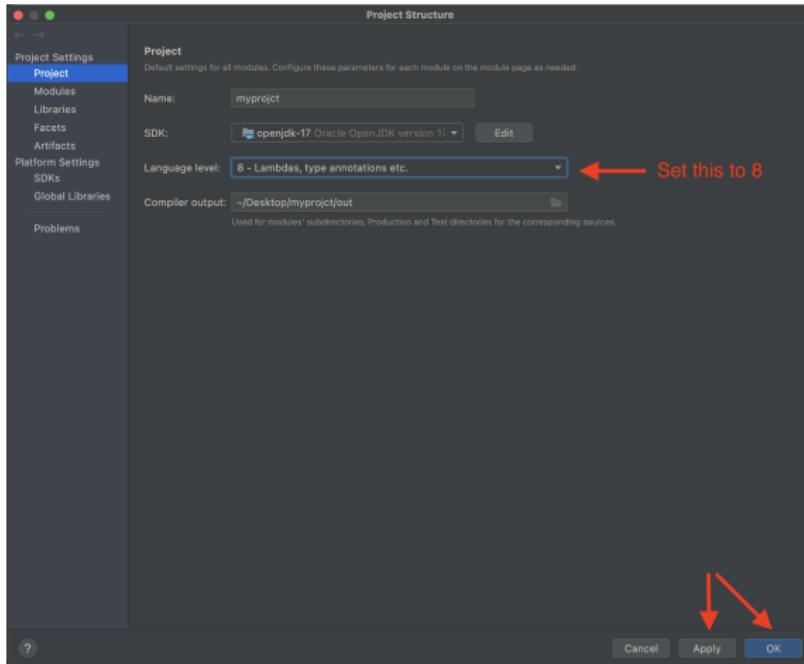
Step 4: Select the Java SDK (CONTINUED)

In the next dialog, select the "Project" settings in the left sidebar. If the Project SDK is set to "No SDK", use the dropdown box to select a valid SDK. Any version 1.8 or higher should be fine. If you don't see a valid SDK, you'll need to install one: Choose "Add SDK" then "Download JDK".



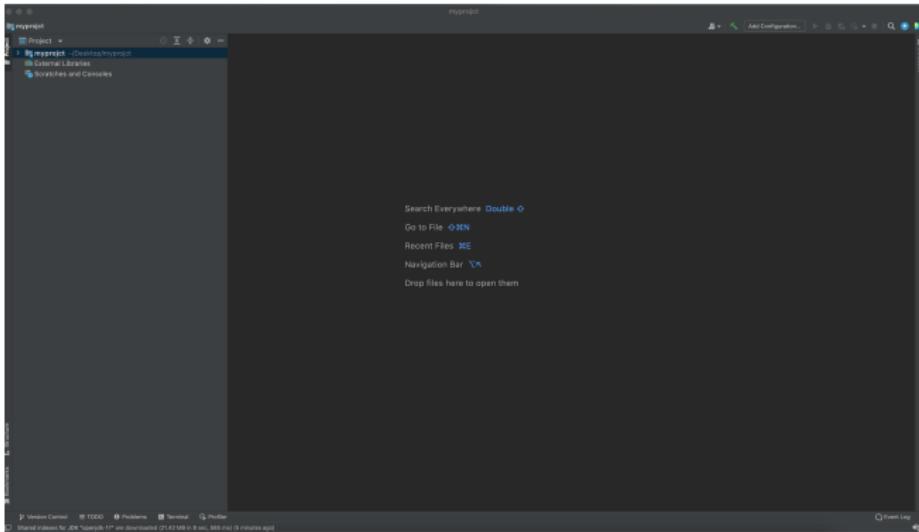
Step 4: Select the Java SDK (CONTINUED)

Set the language level to 8. We don't use any features above language level 8 in CMPT 280. Then click "Apply" and "OK".x



Step 5: Finish

You should now see your empty project. It is now ready for you to add modules. You'll be able to do all your programming for the whole course within this IntelliJ project.



Part II

Adding lib280 as a module of your
project.

Importing lib280 into your project.

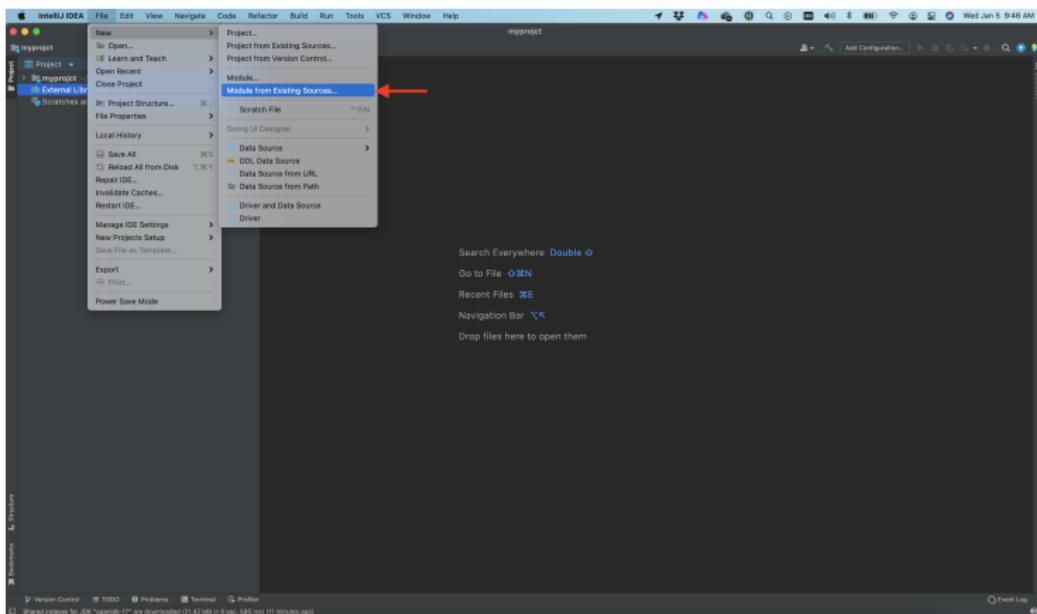
- With each assignment you'll be given a new version of lib280 to use.
- The following slides show how to import a version of lib280 into your IntelliJ project as a module.
- You'll need to do this once for each assignment.

Step 1: Download lib280 from the Course Website

- Download a `lib280-asnX.zip` file from the course website.
- Unzip it and store it somewhere where it won't be accidentally deleted (important!).
- Do **not** unzip these files into your project folder.

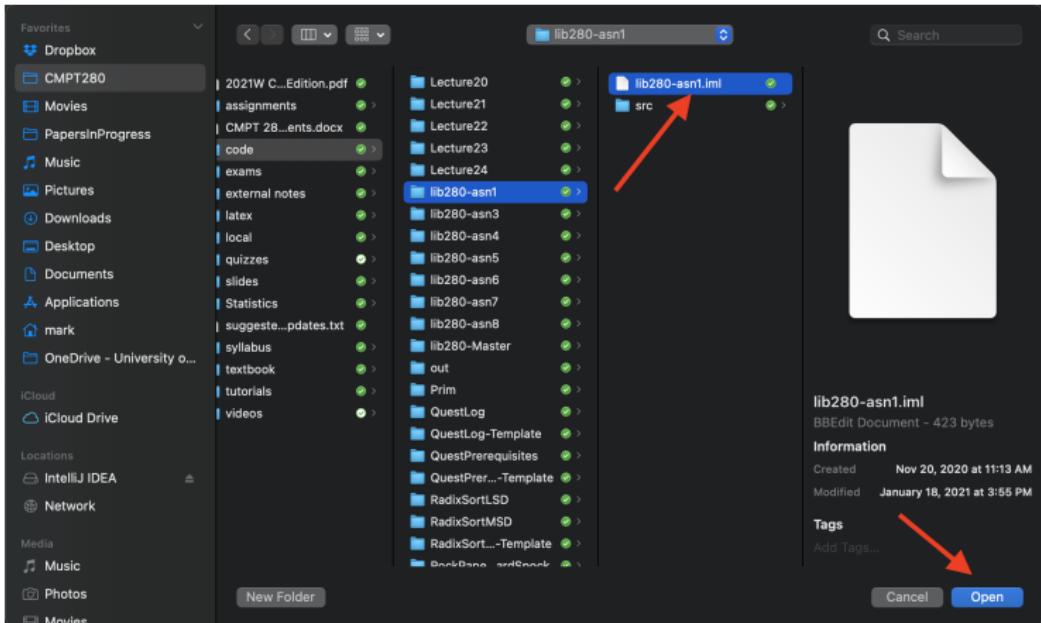
Step 2: Add a Module

From the "File" Menu, choose "New" and then "Module from Existing Sources".



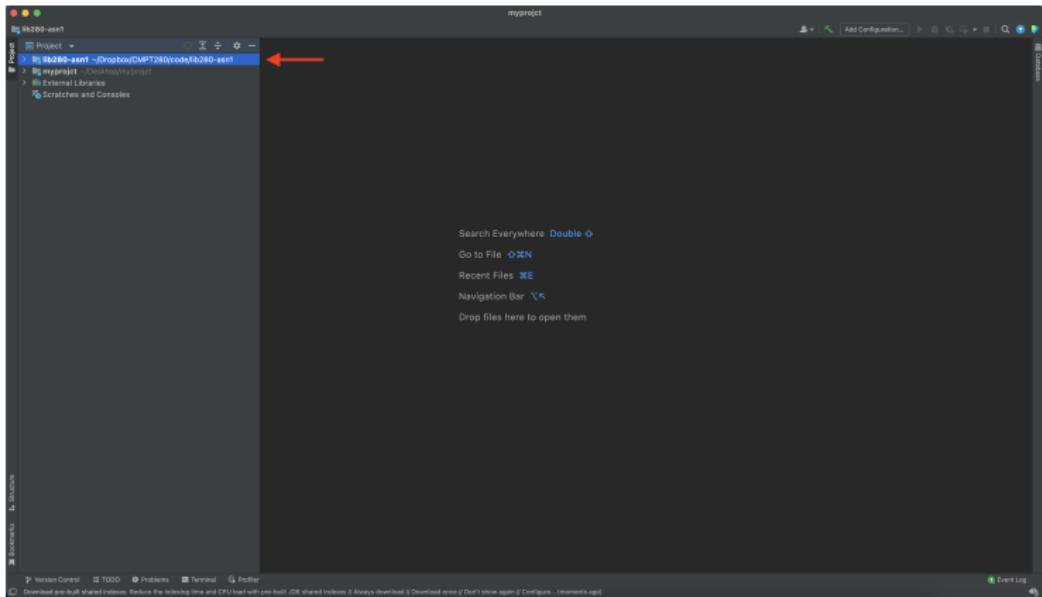
Step 3: Select The Folder Containing Lib280

In the file selection dialog, navigate to the folder containing the files in the lib280 archive you unzipped in Step 1 and select the **.iml file**. Click "Open".



Step 4: Module is Created.

You'll now have a module containing the lib280 code for the given assignment.



Part III

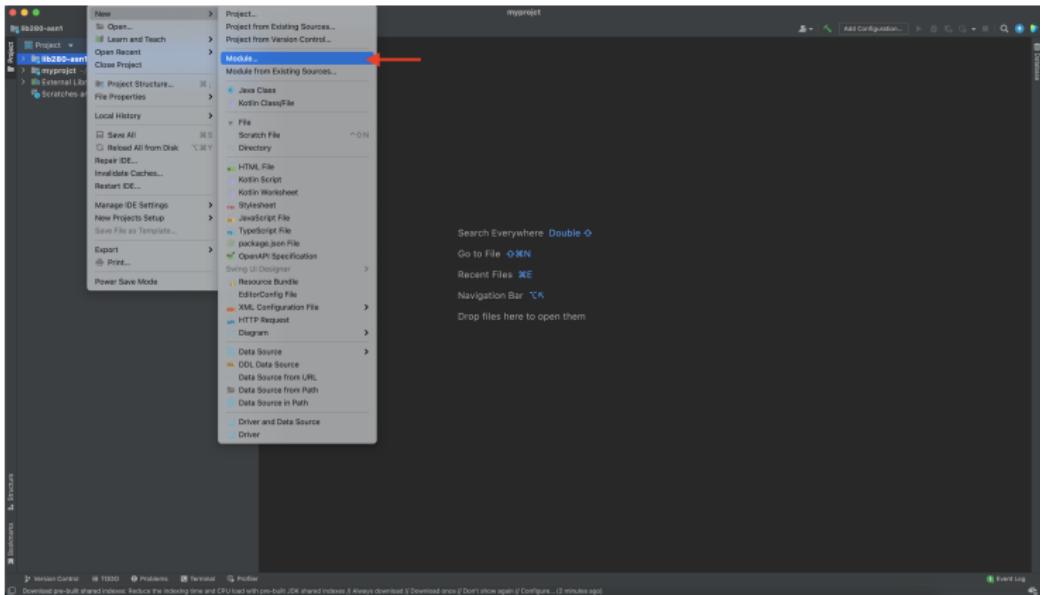
Creating a program that uses a lib280
library.

Creating a program that uses lib280.

- Some assignment questions will only require that you modify code within, or add code to lib280. If that's the case, then you do not need to follow the steps in this part.
- Other assignment questions will require that you write a separate program that **uses** the classes in lib280. If you want to do that, then you need to create a new module that has a lib280 library module as a *dependency*.
- You'll need to do this for each **assignment question** that requires a program that is separate program from lib280.

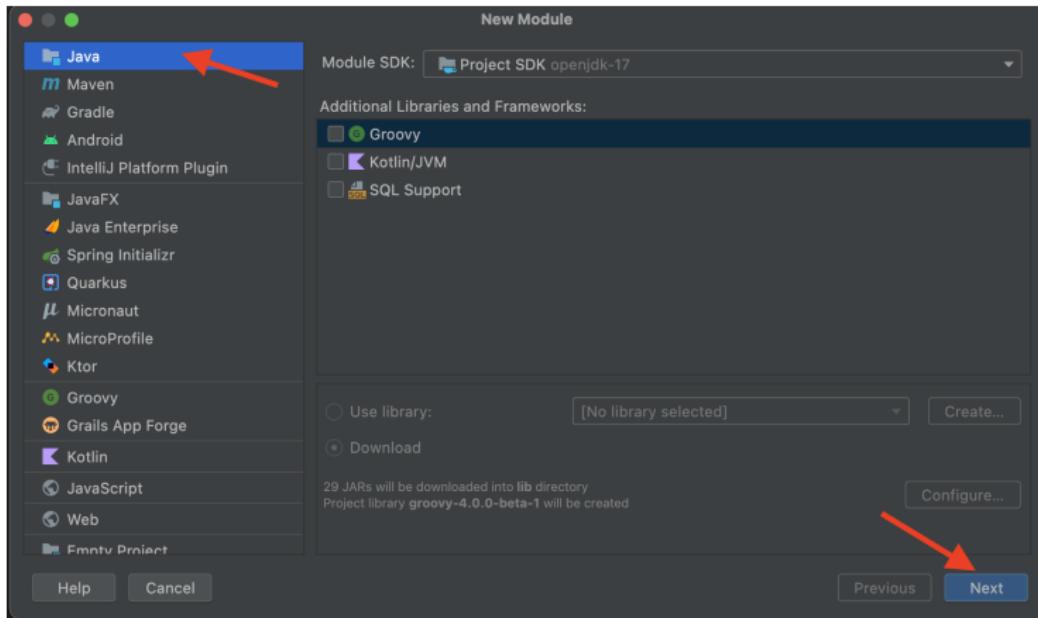
Step 1: Create a new Module

From the "File" menu, choose "New" and then "Module".



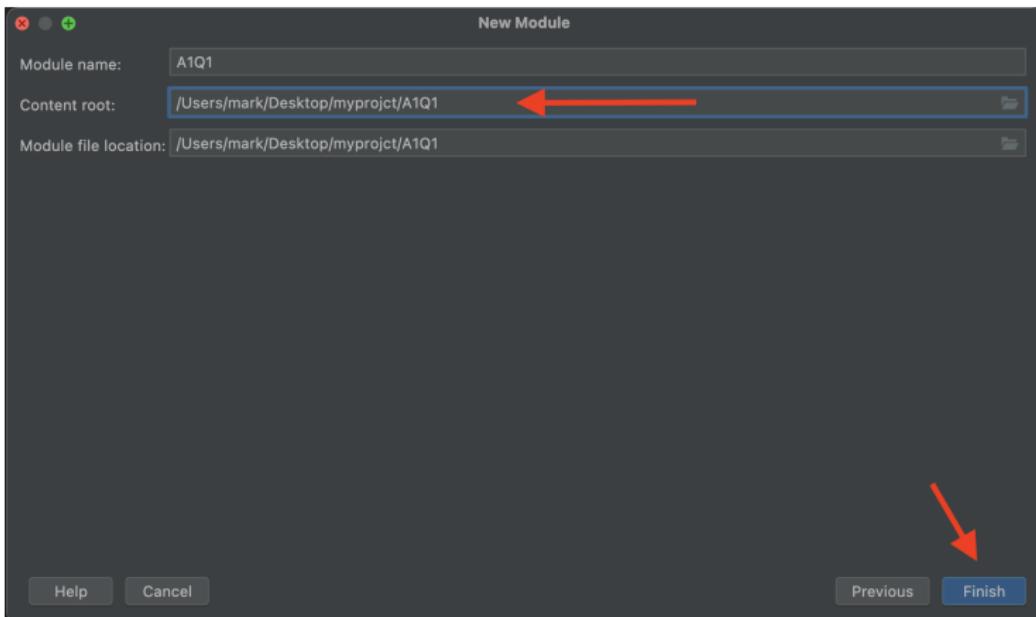
Step 2: Choose Module Type

Make sure "Java" is selected on the left, and then click "Next".
Don't check any of the checkboxes.



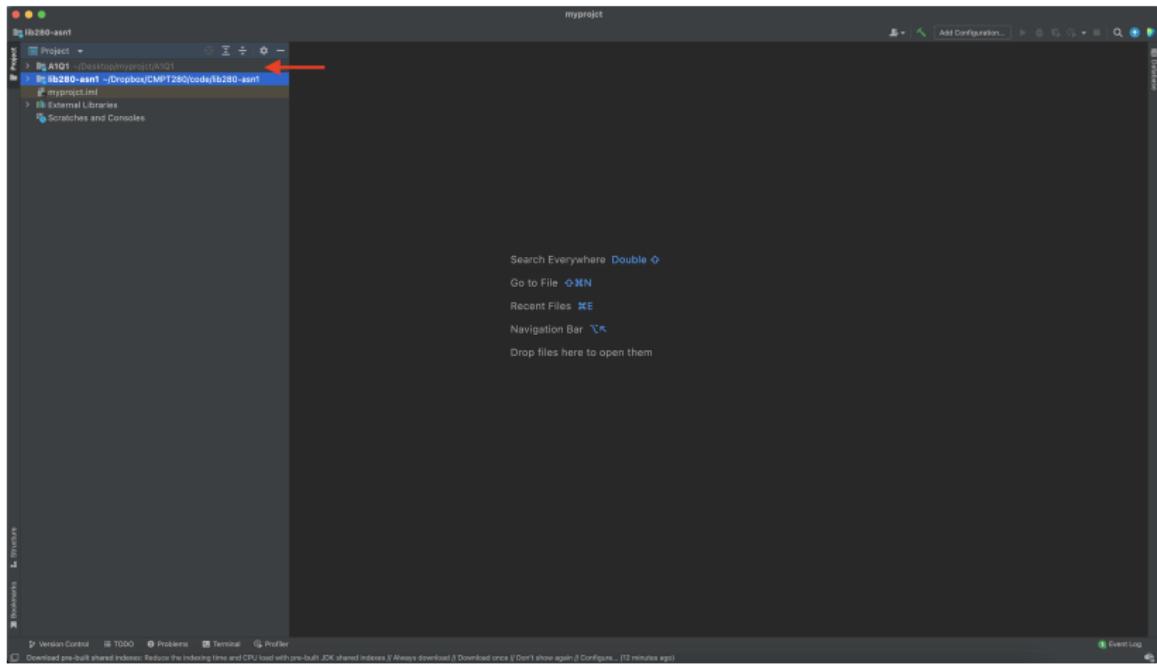
Step 3: Name your Module

In the next dialog, set the “content root” to be a sub-folder of the main project folder that you created earlier (be careful here because the default value may not be your project folder). Then make sure the “module file location” is the same as the “content root”, and make sure the “module name” is the same as last folder name in the “content root”.



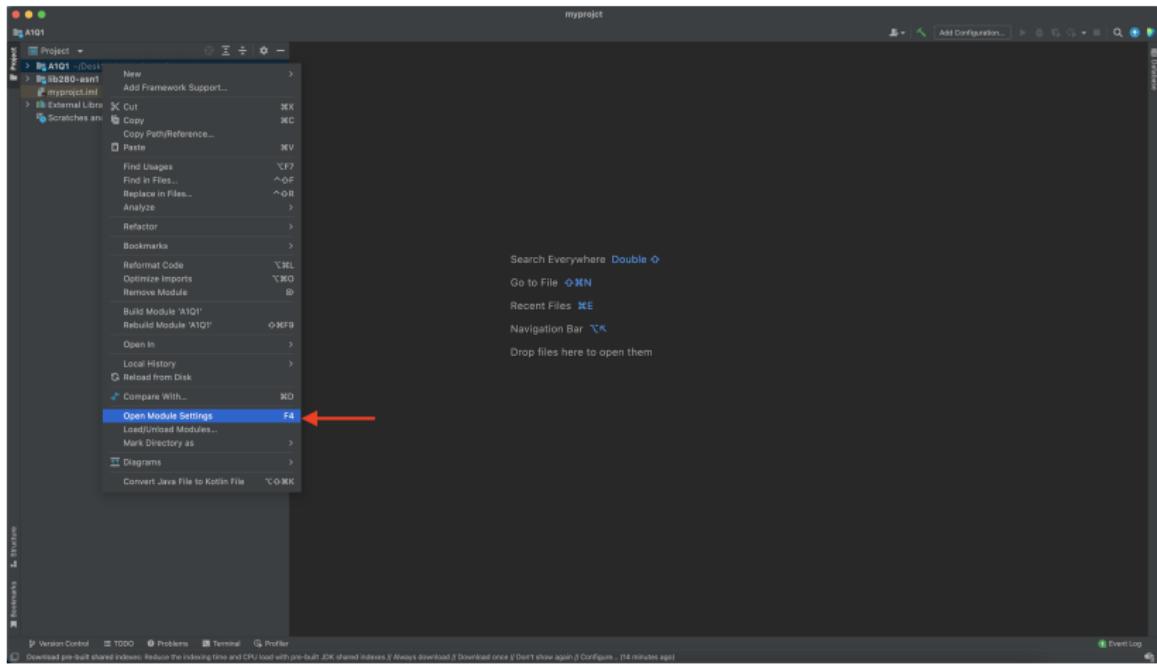
Step 4: Module is Added

You'll now see that the module has been added.
But now we have to grant it access to lib280...



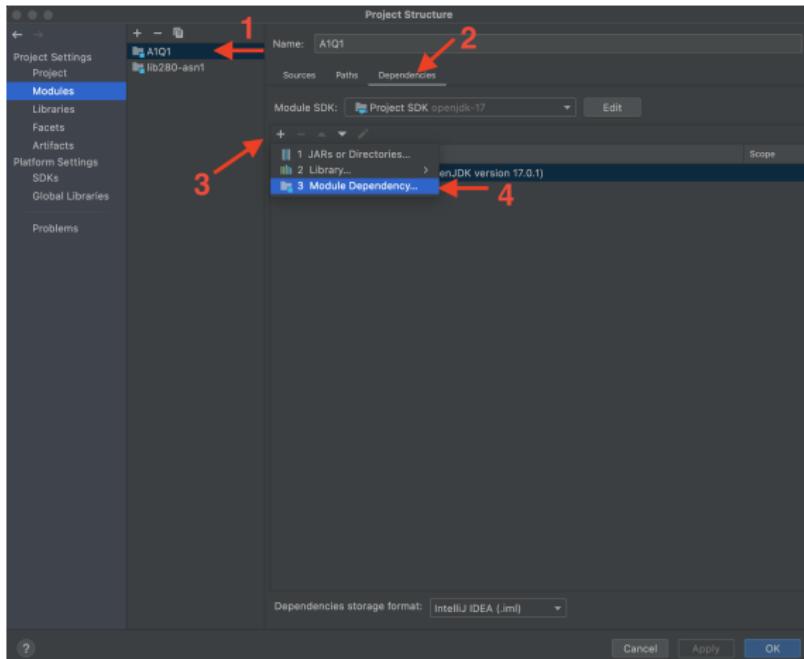
Step 5: Add module dependency

Right-click on the module you just added. Choose "Open Module Settings".



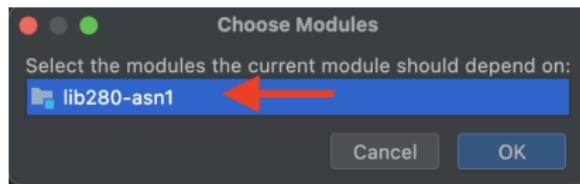
Step 6: Add module dependency

Make sure the newly added module is selected on the left. Click the "Dependencies" tab in the main pane. Click the "+" button, at the top of the tab, then select "Module Dependency".



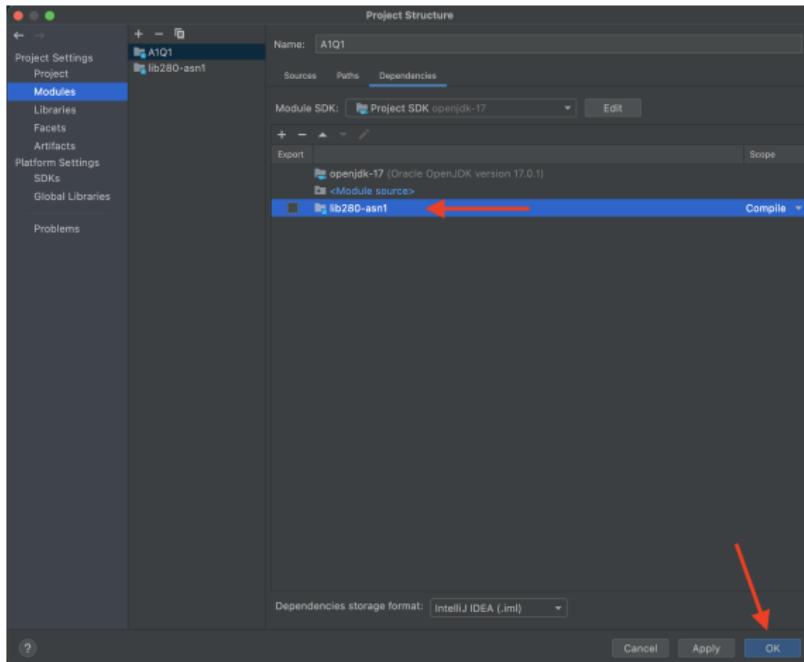
Step 7: Select the lib280 Module to add as Dependency

Select the lib280 module you want your program to have access to.
Make sure you pick the right one once you have more than one!
Click "OK".



Step 8: Verify Dependency

Verify that the correct module has been added as a dependency and click "OK".



Step 9: Add classes and start coding.

Your module is now ready. You can now add a class to the module by right-clicking on the module's "src" folder as shown below. You'll be able to import lib280 classes into the newly added class.

