Meme Credit: https://dzone.com/articles/intellij-idea-vs-eclipse-which-is-better-for-begin

Lab 1: Getting Started

CS 2112 Fall 2025

August 25 / August 27, 2025

Lab Staff

Monday

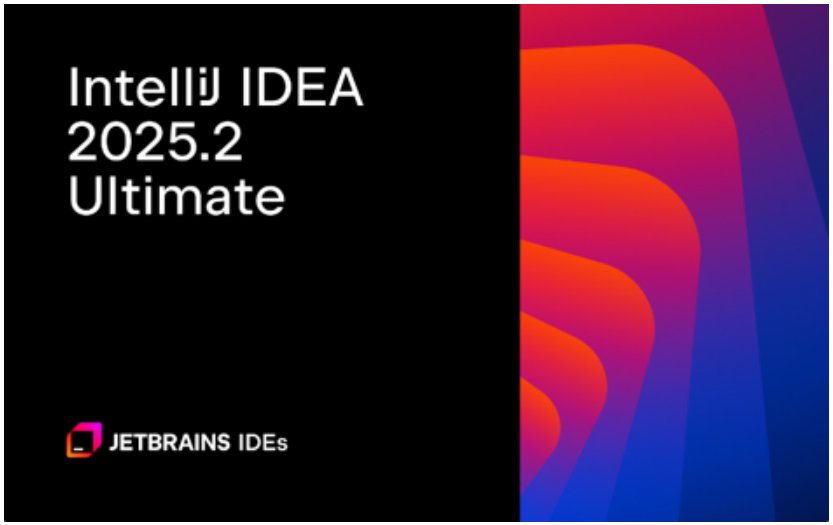
Mikael Andreasyan
Arnab Ghosh
Arin Khare
Andrew Kim
Michael Xing

Wednesday

Arnab Ghosh
Matthew Kim
Riddhi Lamba
Michael Xing
Shawn Zou

Versions

We recommend IntelliJ
(any relatively new version should be acceptable)



Community should be fine for our needs, but you can download Ultimate for free with your Cornell email.

We require Java 21.

Can I Use Another IDE?

Yes, but we will not be able to provide any technical support.
If something breaks, you're on your own.

Step 0 - Install IntelliJ

Go to

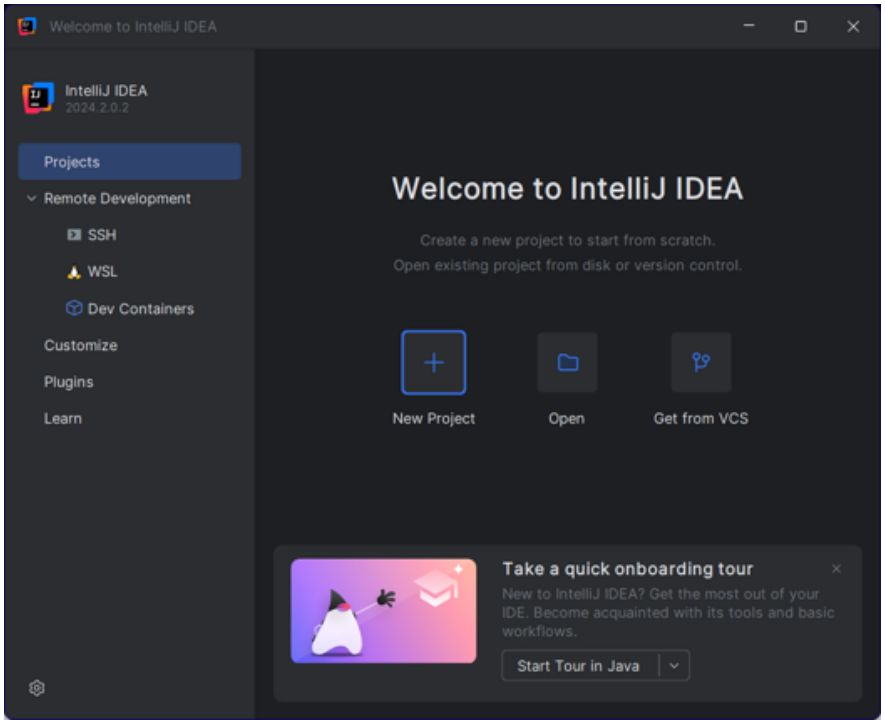
https://www.jetbrains.com/idea/download/

Install your choice of Community Edition or Ultimate (using your student email)

Linux: Instructions vary depending on distribution. Check your package manager of choice.

Step 1 - First IntelliJ Run

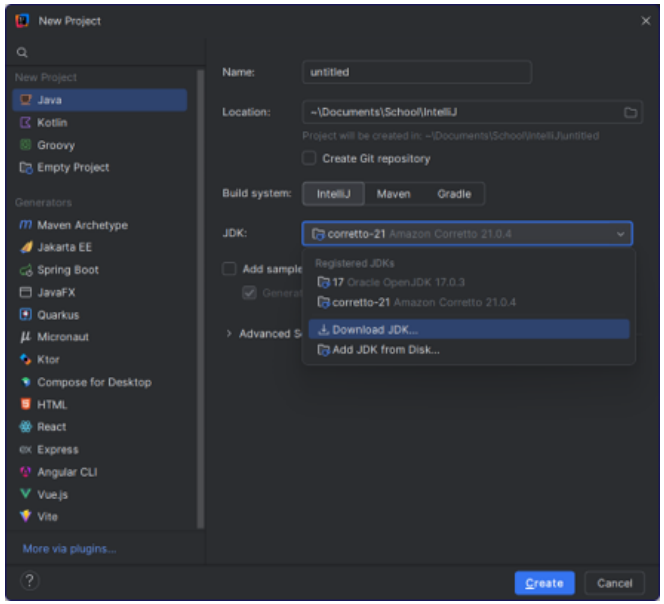
You should come to a screen like this:



Click ‘New Project’

Step 2 - New Project

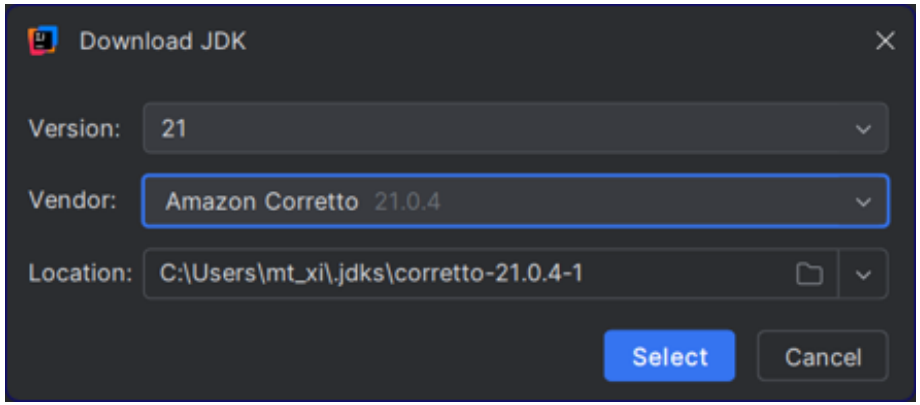
You should see this:



Under the "JDK" dropdown, if you do not see an installation of JDK 21 already, choose "Download JDK" to get one.

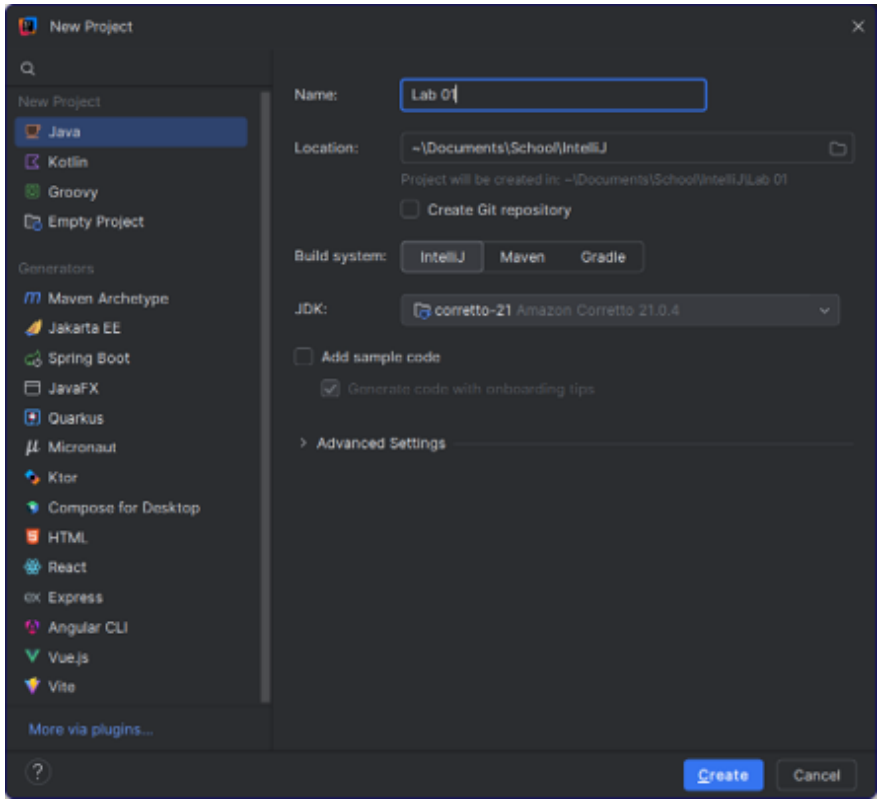
Step 3 - Download JDK

Choose version "21" and vendor "Amazon Corretto"



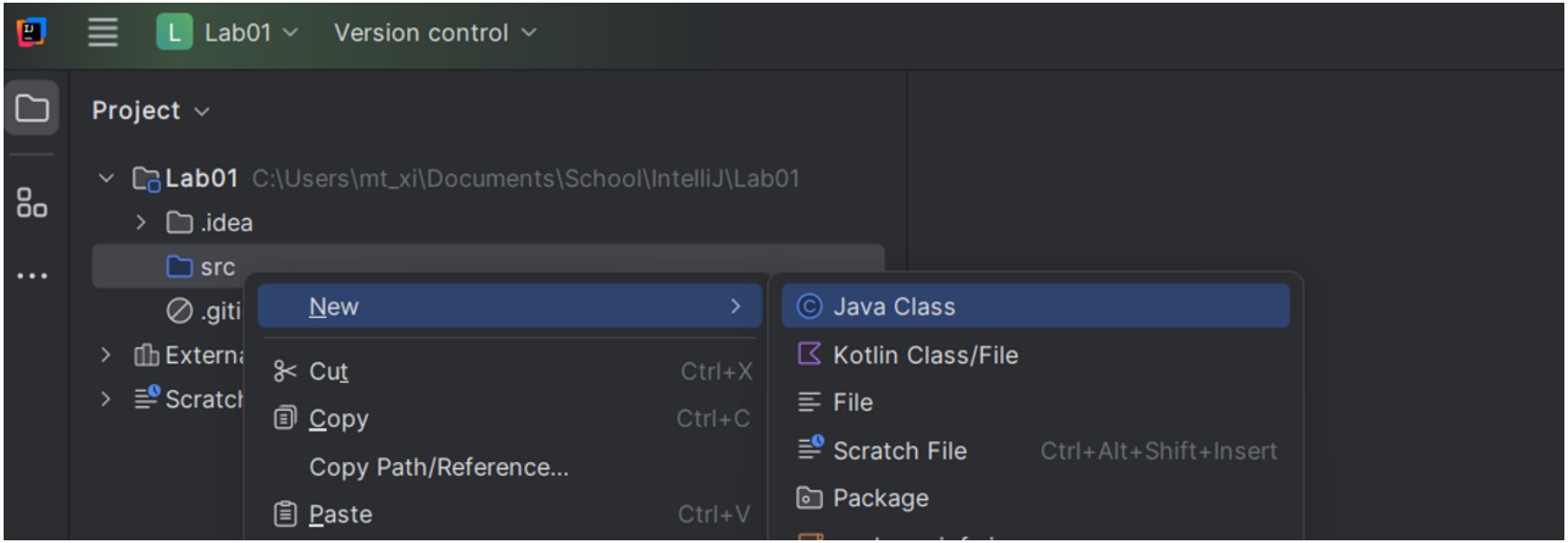
Step 4 - New Project (for real)

Name your project and make sure you **choose JDK 21**.
Uncheck "Add sample code"



Step 5 - Create Class

Right click `src` and choose New → Java Class.



Enter a name for the class (like "`Hello`") and press Enter.

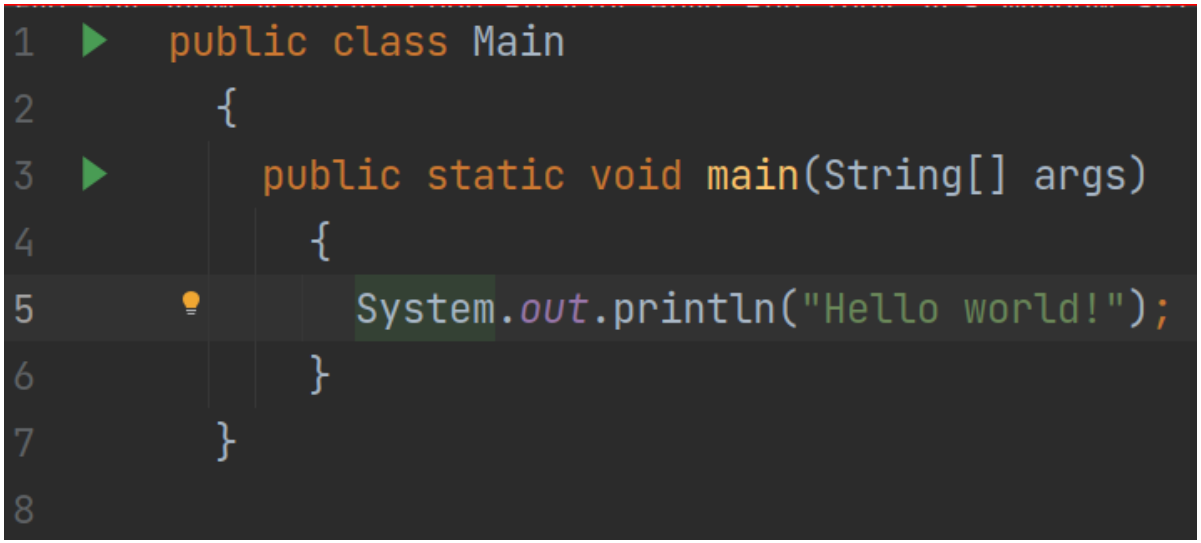
Hello World

Type "psvm<TAB>" to autocomplete a boilerplate main method.

```
1 public static void main(String[] args) {
2     System.out.println("Hello World");
3 }
```

Running your code

Click one of the green triangles on the sidebar



After the first run, you can also use the Run/Debug toolbar on the top right

Coding Exercise

There is a 4-digit number which, when the order of its digits is reversed, yields a number 4 times greater.

$$dcba = 4 * abcd$$

$$a \neq 0$$

Write a program to find and print out this number.

Useful Features Of IntelliJ

- Autosave
  - Should be on by default
  - Can manually save with Ctrl + S or Cmd + S
- Autocomplete
- Autoindent
- Compile & Run
- Refactoring
- Autoformat (Ctrl + Alt + L/Opt + Cmd + L)
- Javadoc
- Comments

Autoformatting

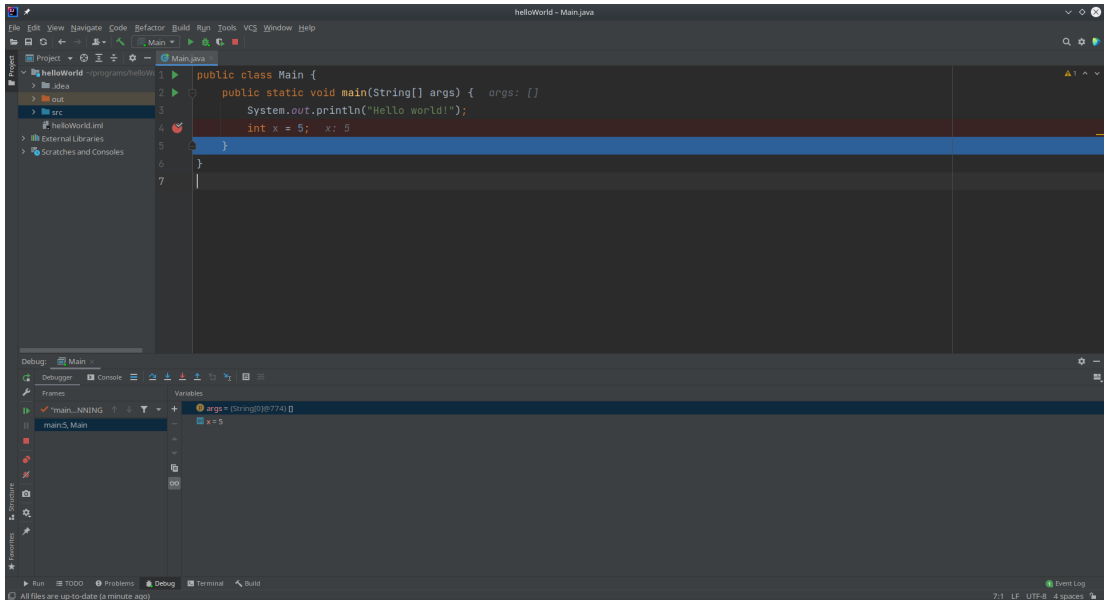
- Ctrl + Alt + L
- Opt + Cmd + L
- We suggest turning on autoformat on save in File > Settings > Tools > Actions on Save > check "Reformat Code"
- Once you have a Git repo, pick "Changed lines" instead of "Whole file" (wait for Lab 5)
- If your group members don't have the same format, you will get terrible merge conflicts

Refactoring Tools

- Automatically change names of variables, classes, methods, etc.
- Right Click > Refactor > Rename

Debug View

You can enter debug view by clicking the bug icon next to Run.
Debug view allows you to add breakpoints, step through individual lines of code, inspect variable contents, and test expressions to see their results.
To stop debugging, click on the red square on the left of the debug window.



Using Debug View

Double click to the left of a line number to set or unset a breakpoint. Execution will pause when it reaches a breakpoint.

Use the "Variables" pane to inspect the values of variables in scope.

Use the step buttons at the top of the debug pane to move through your code. Step Into will go into the highlighted method call, Step Over will run the selected method and pause on the next line, and Step Out takes you to the caller of the current method.

Use the + button at the left of the "Variables" pane to test the value of various expressions.

Breaking on Exceptions

Choose Run > View Breakpoints > Java Exception Breakpoints to add a breakpoint whenever an exception is about to be thrown.

This allows you to use the debugger to track down the source of exceptions

Debugging Exercise

Linked on the course website under today's lab is an exercise file with code for the previous exercise.

It doesn't work.

Using the debugger, try to find out where and why it's broken.