



# Data Structure

**Lab Session:  
Introduction**

**U Kang  
Seoul National University**



# Goals

- Goals in the lab sessions
  - How to use Java language
  - How to write programs using data structures
  - How to prepare programming assignments
- You can easily finish the assignments if you participate in the sessions actively
- Lab Sessions will be proceeded in English but we will receive questions both in English and Korean so feel free to ask!

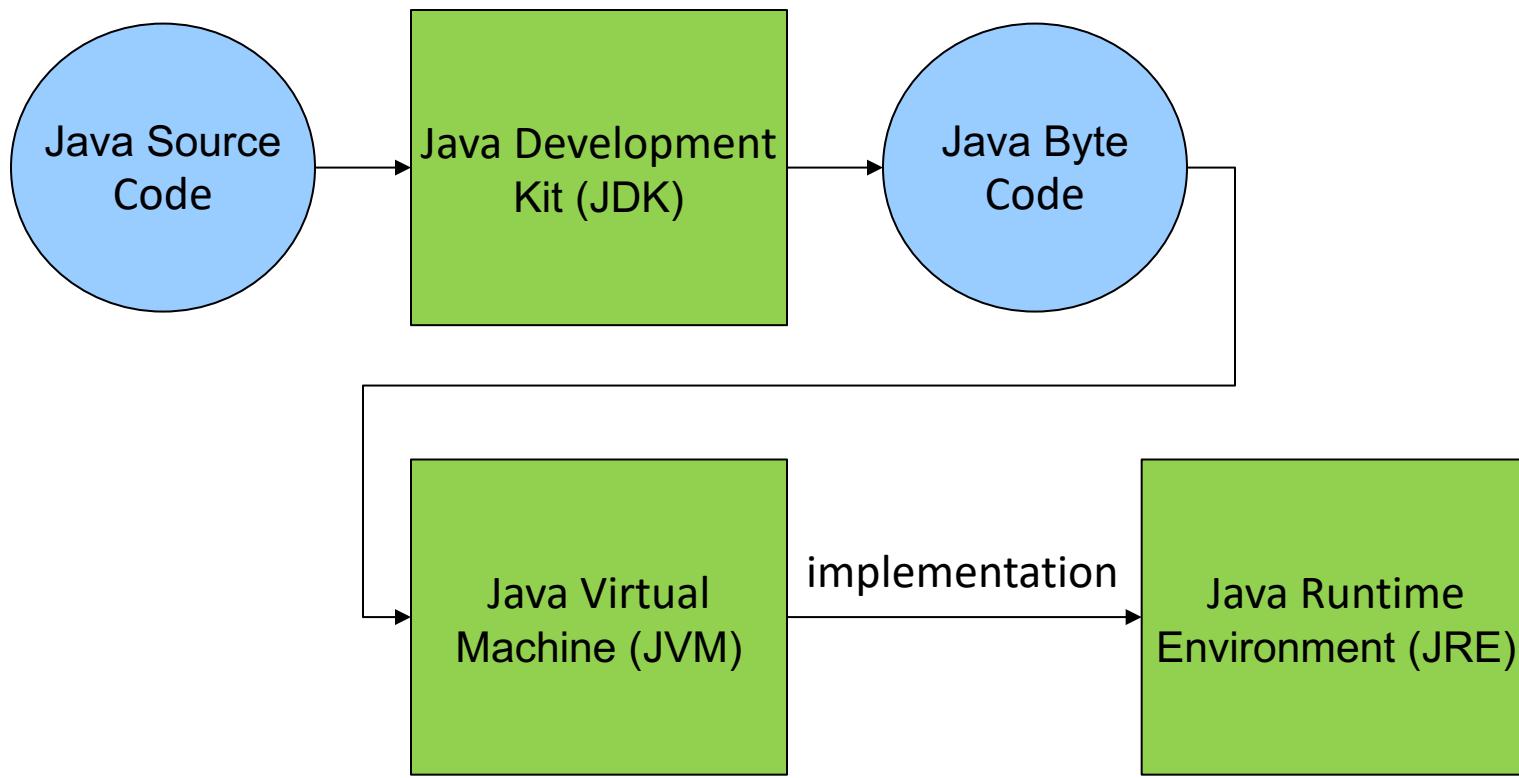


# How Java Works?

- JDK (Java Development Kit)
  - JDK is a bundle of software components that is used to develop Java based applications
- JVM (Java Virtual Machine)
  - JVM executes pre-compiled Java programs
- JRE (Java Runtime Environment)
  - JRE is an implementation of the JVM which actually executes Java programs



# How Java Works? (cont.)





# Install JDK

- We will use Java JDK 11 for our course
- <https://www.oracle.com/technetwork/java/javase/downloads/index.html>
- Click Download

## Java SE 11.0.4 (LTS)

Java SE 11.0.4 is the latest release for the Java SE 11 Platform

[Learn more ➔](#)

- Installation Instructions
- Release Notes
- Oracle JDK License
- Java SE Licensing Information User Manual
  - Includes Third Party Licenses
- Certified System Configurations
- Readme

## Oracle JDK

[DOWNLOAD ➔](#)



# Install JDK

- Accept License
- (Window) jdk-11.0.4\_windows-x64\_bin.exe
- (Mac OS) jdk-11.0.4\_osx-x64\_bin.dmg

**Java SE Development Kit 11.0.4**

You must accept the [Oracle Technology Network License Agreement for Oracle Java SE](#) to download this software.

→  Accept License Agreement  Decline License Agreement

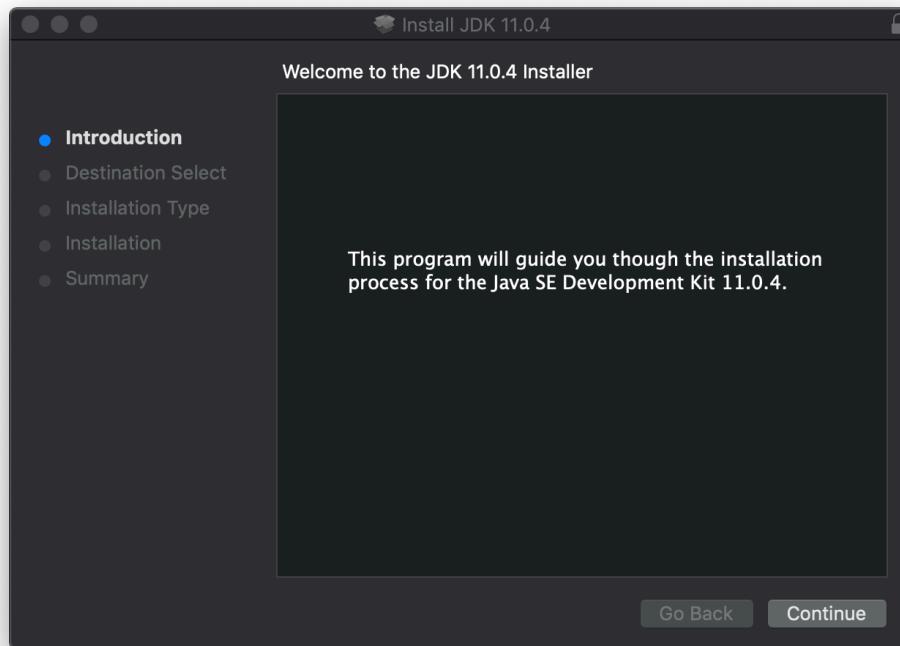
Product / File Description	File Size	Download
Linux	147.85 MB	<a href="#">jdk-11.0.4_linux-x64_bin.deb</a>
Linux	154.6 MB	<a href="#">jdk-11.0.4_linux-x64_bin.rpm</a>
Linux	172.01 MB	<a href="#">jdk-11.0.4_linux-x64_bin.tar.gz</a>
macOS	166.58 MB	<a href="#">jdk-11.0.4_osx-x64_bin.dmg</a> ←
macOS	166.95 MB	<a href="#">jdk-11.0.4_osx-x64_bin.tar.gz</a>
Solaris SPARC	188.21 MB	<a href="#">jdk-11.0.4_solaris-sparcv9_bin.tar.gz</a>
Windows	151.22 MB	<a href="#">jdk-11.0.4_windows-x64_bin.exe</a> ←
Windows	171.25 MB	<a href="#">jdk-11.0.4_windows-x64_bin.zip</a>



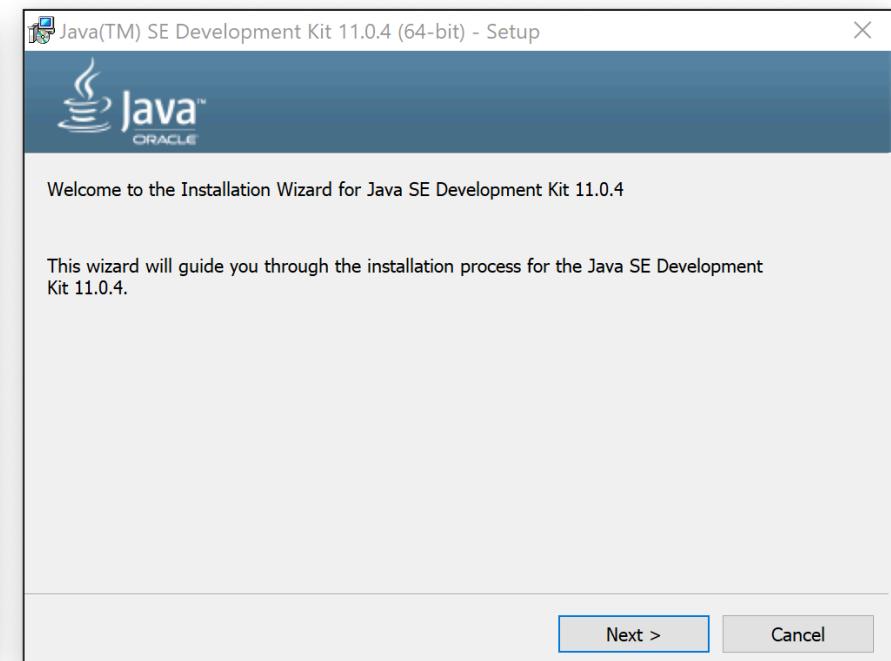
# Install JDK

- Click Continue or Next

Mac OS



Window

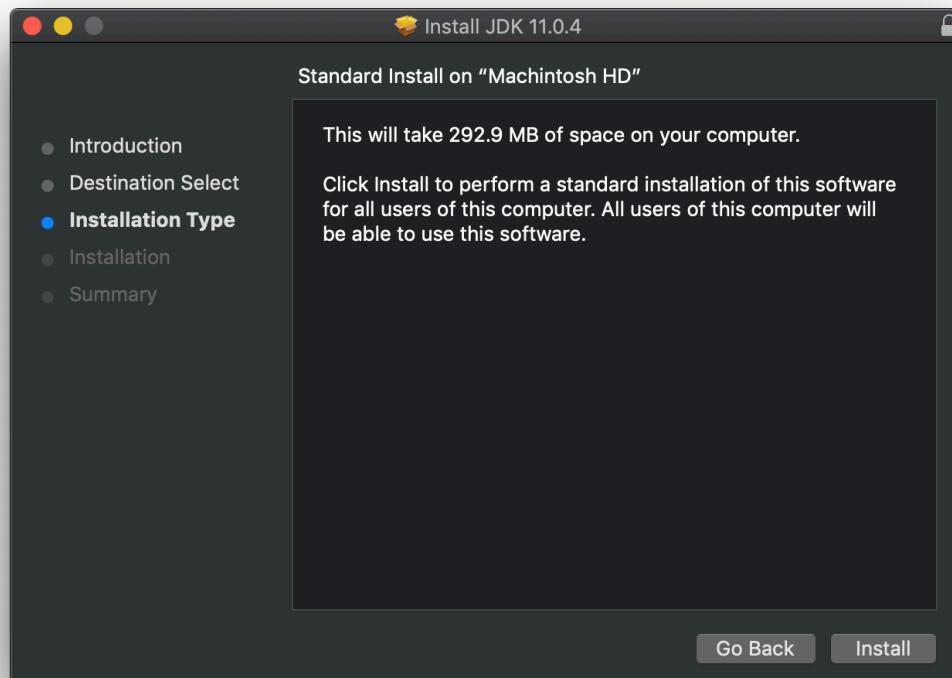




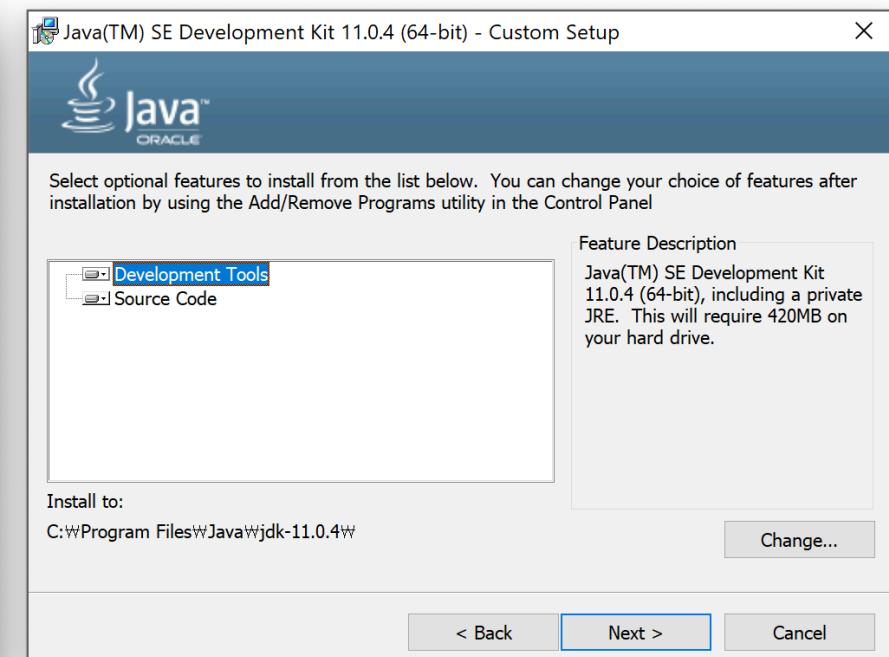
# Install JDK

- Click Install or Next

Mac OS



Window



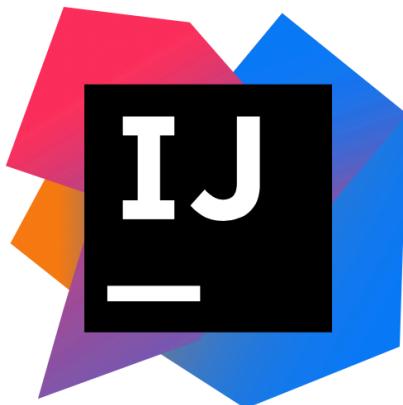


# How to Code?

- Text pad and terminal
- Source-code editor
  - Code editor with syntax highlighting
  - Visual Studio Code, Atom, Sublime Text
- Integrated Development Environment (IDE)
  - Source-code editor + various tools to aid programmer
  - **IntelliJ IDEA**, Eclipse, NetBeans
  - IntelliJ IDEA comes with JDK 11 preinstalled



# Install IntelliJ IDEA



Version: 2019.2.1

Build: 192.6262.58

August 21, 2019

[Release notes](#)

## Download IntelliJ IDEA

[Windows](#)

[macOS](#)

[Linux](#)

### Ultimate

For web and enterprise development

[DOWNLOAD](#)

### Community

For JVM and Android development

[DOWNLOAD](#)

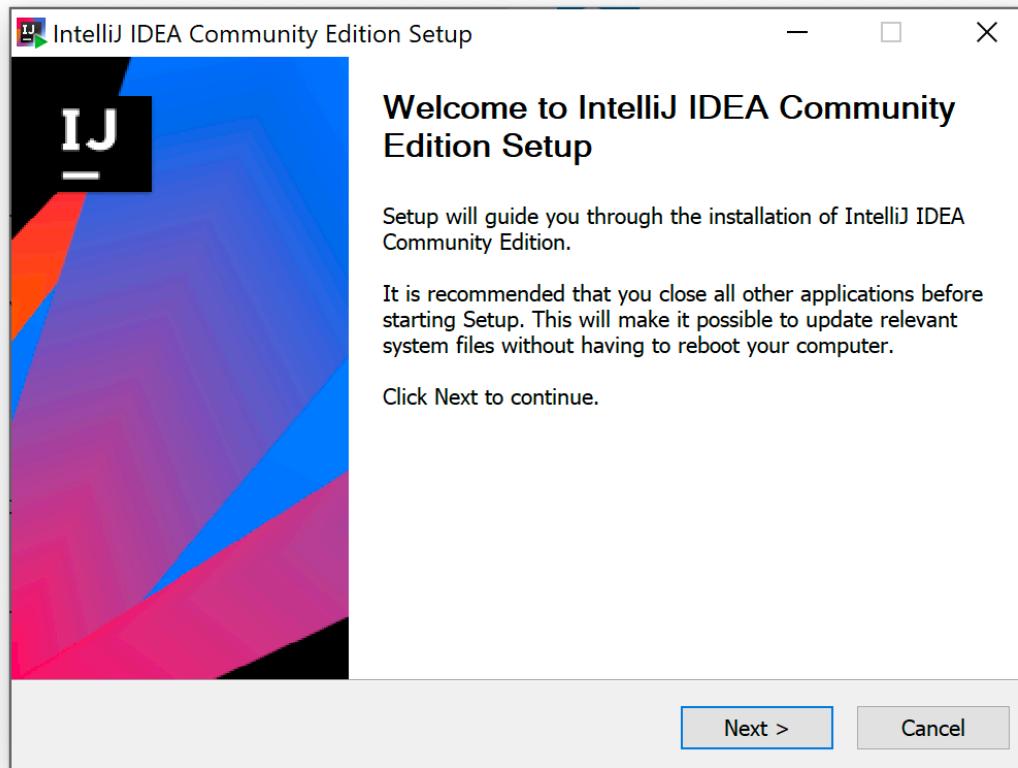


- Install IntelliJ IDEA Community
  - <https://www.jetbrains.com/idea/download>
  - Students who want to use IntelliJ Ultimate can apply for a student pack
    - <https://www.jetbrains.com/student/>
    - For this course, functions in Ultimate version is not required



# Install IntelliJ IDEA (Window)

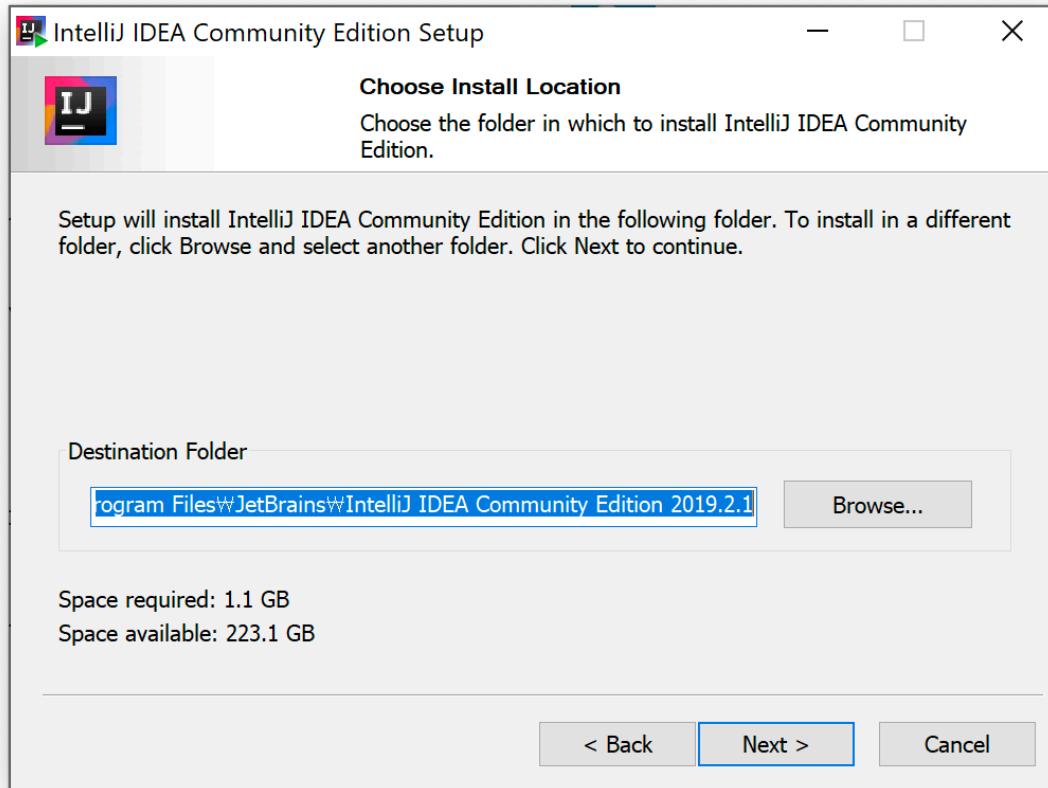
- Open EXE file and click Next





# Install IntelliJ IDEA (Window)

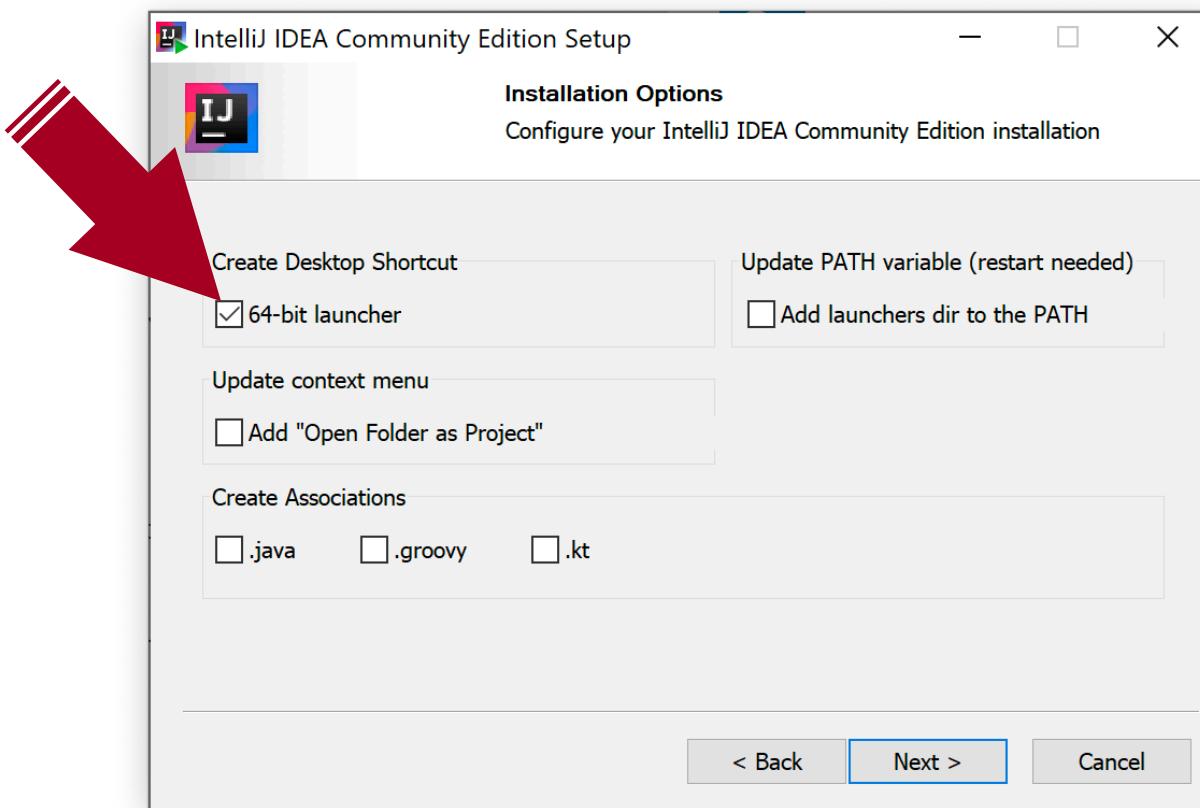
- Click Next





# Install IntelliJ IDEA (Window)

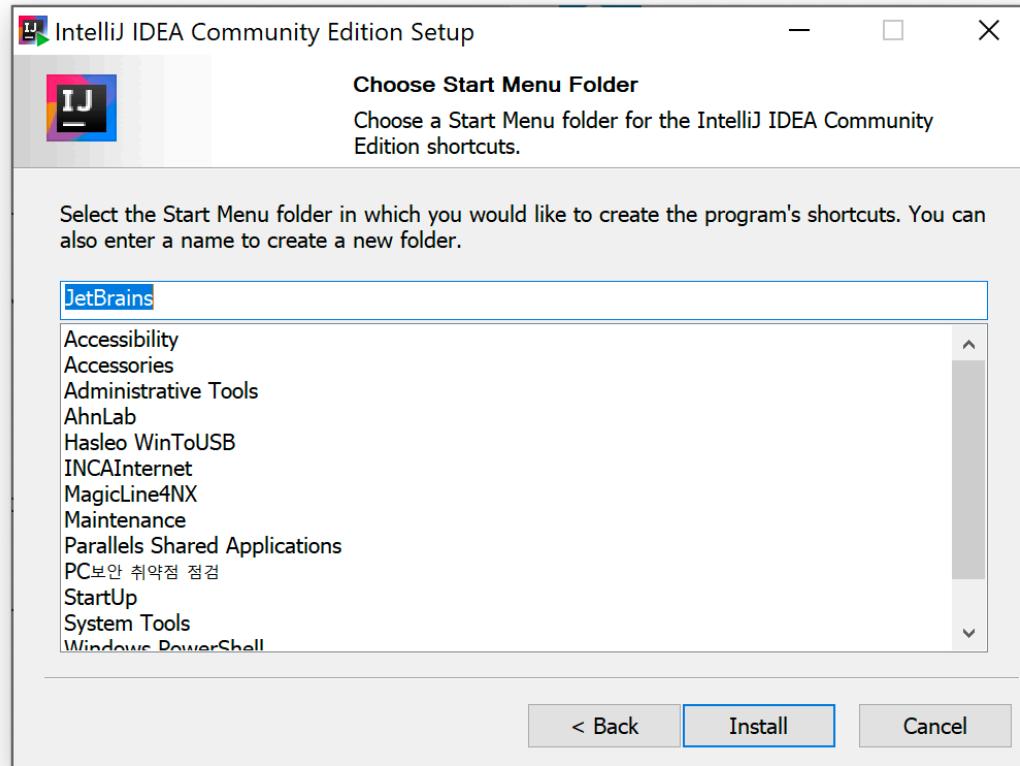
- Click 64-bit launcher and click Next





# Install IntelliJ IDEA (Window)

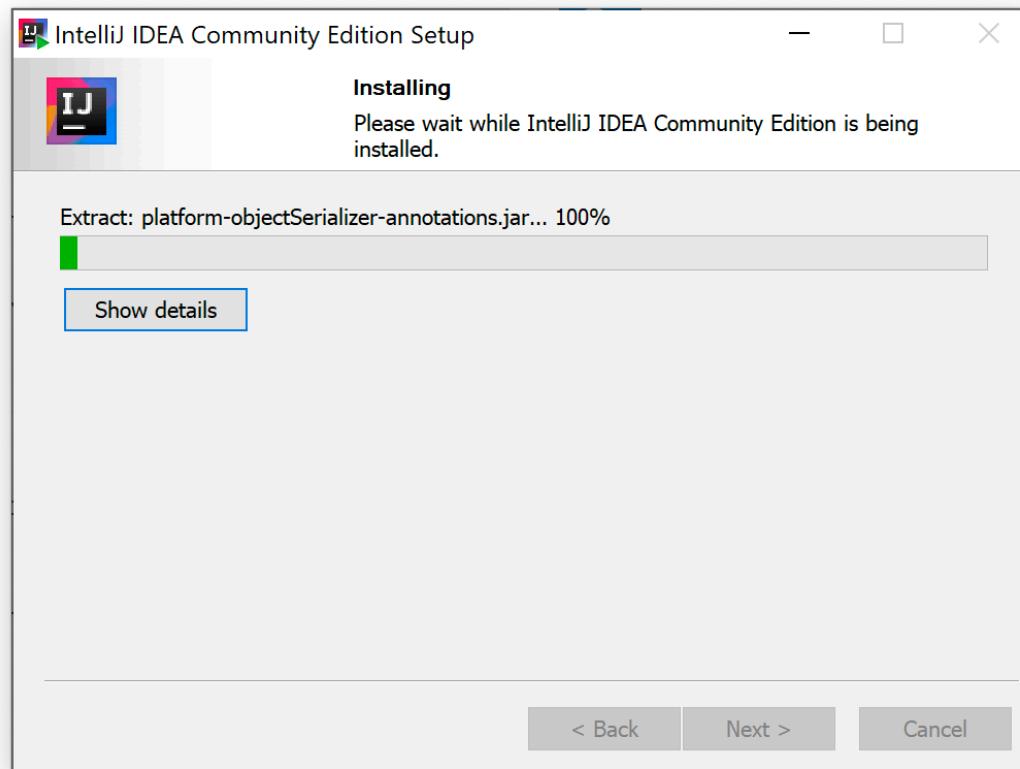
- Click Install





# Install IntelliJ IDEA (Window)

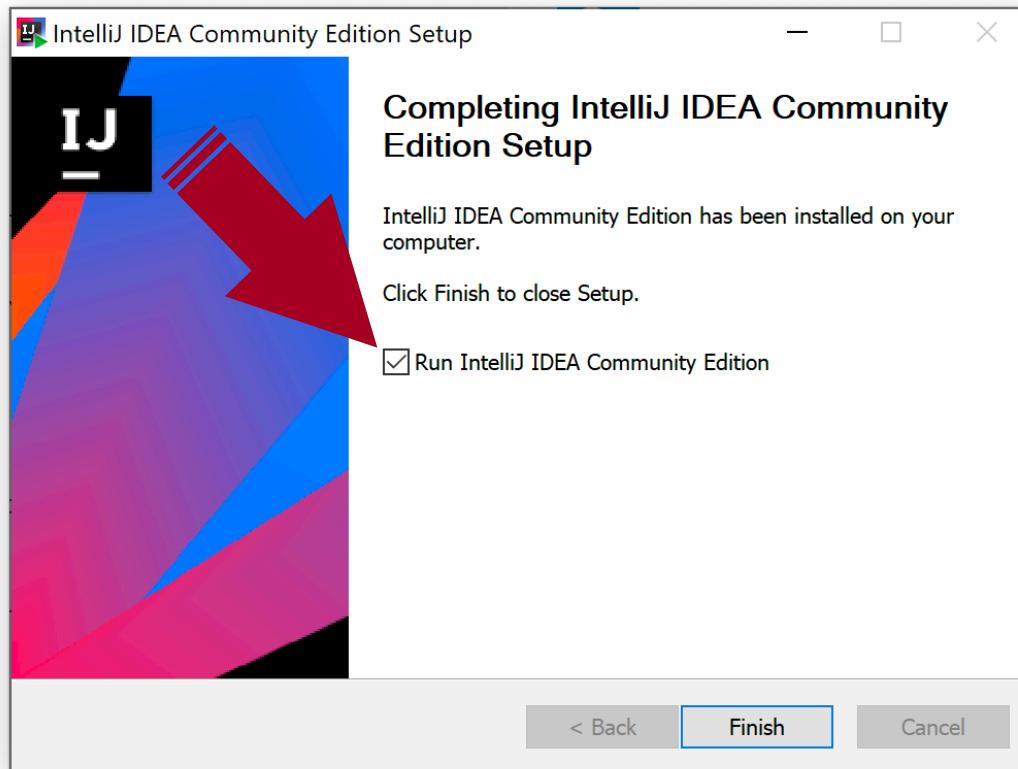
- Wait for it to install





# Install IntelliJ IDEA (Window)

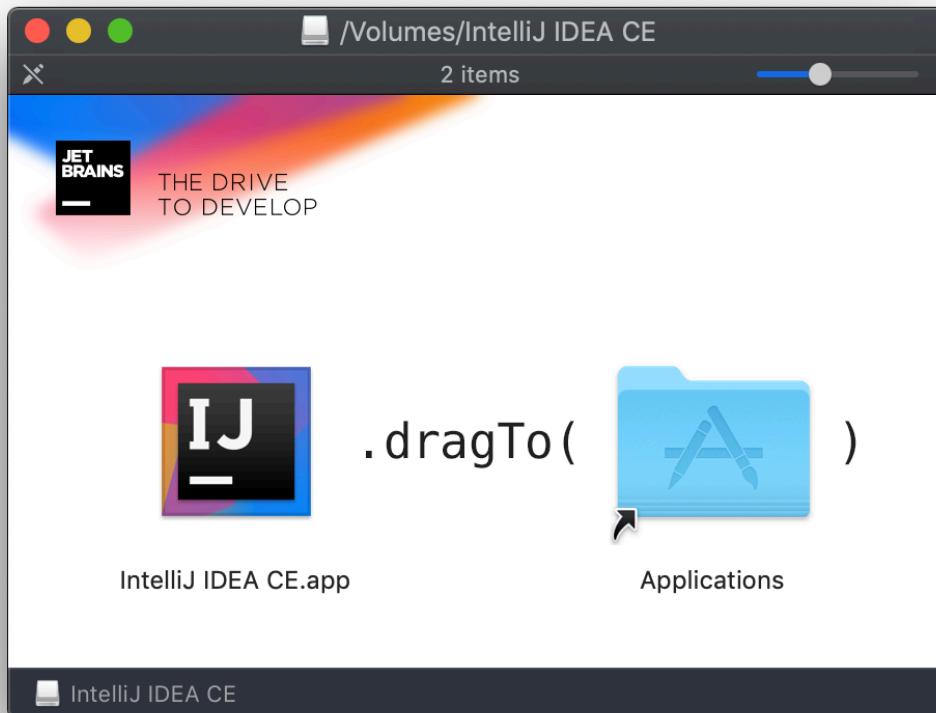
- Click Run IntelliJ IDEA Community Edition and click Finish





# Install IntelliJ IDEA (Mac OS)

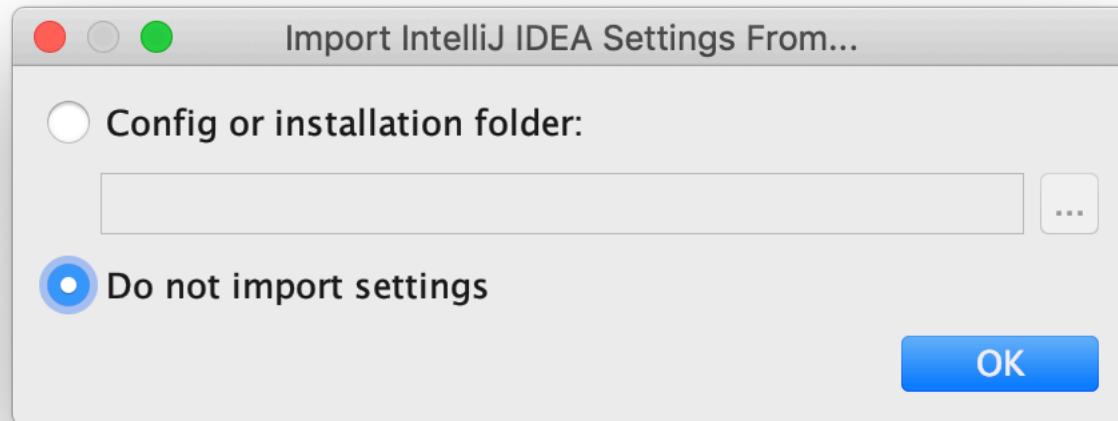
- Open DMG file and drag to Applications folder





# Install IntelliJ IDEA (Mac OS)

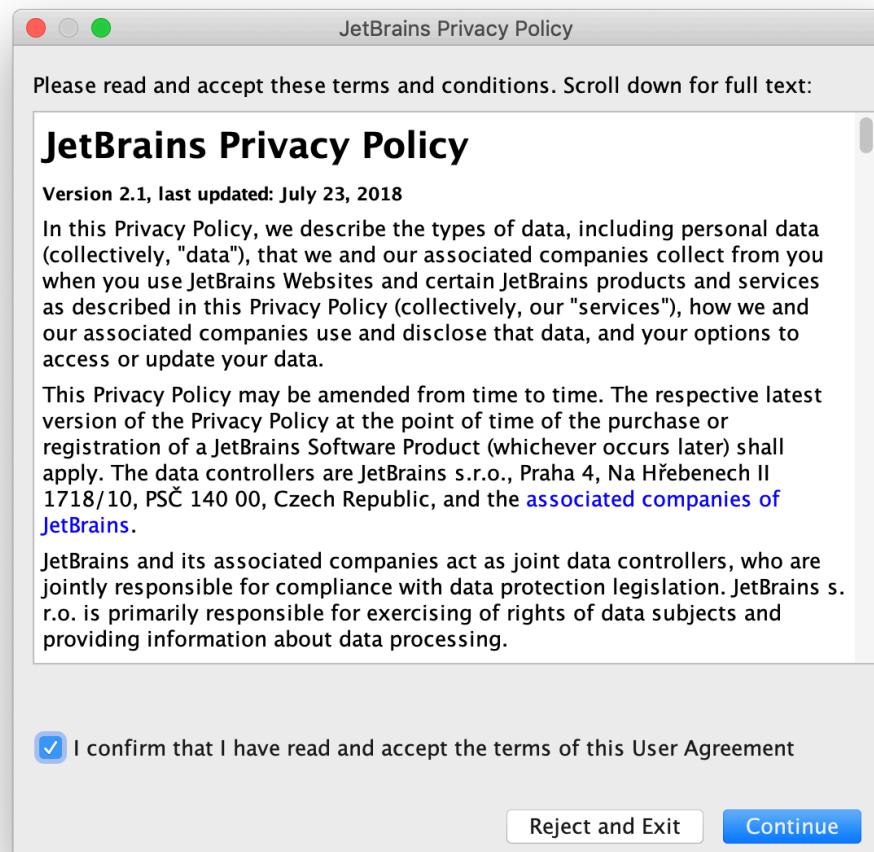
- Open IntelliJ Application and click Do not import settings and press OK





# Install IntelliJ IDEA (Mac OS)

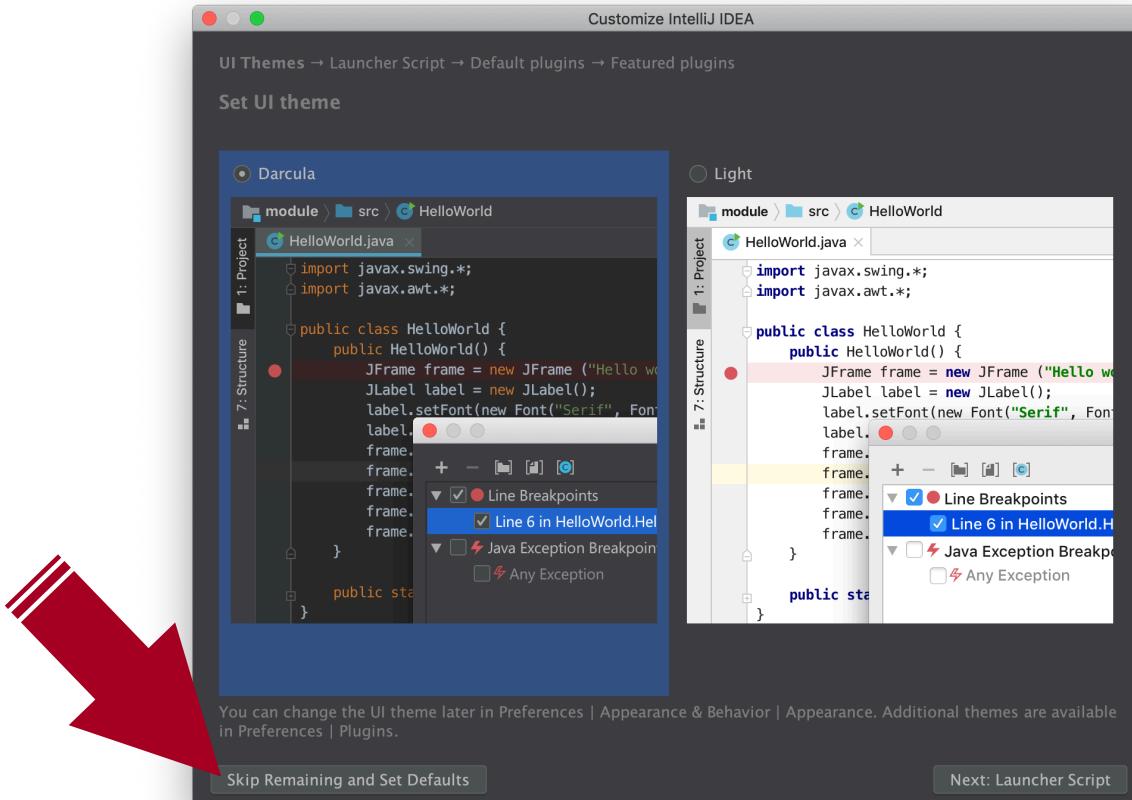
## ■ Confirm Privacy Policy





# Install IntelliJ IDEA (Mac OS)

- Pick a UI theme of your choice and click on Skip Remaining and Set Defaults





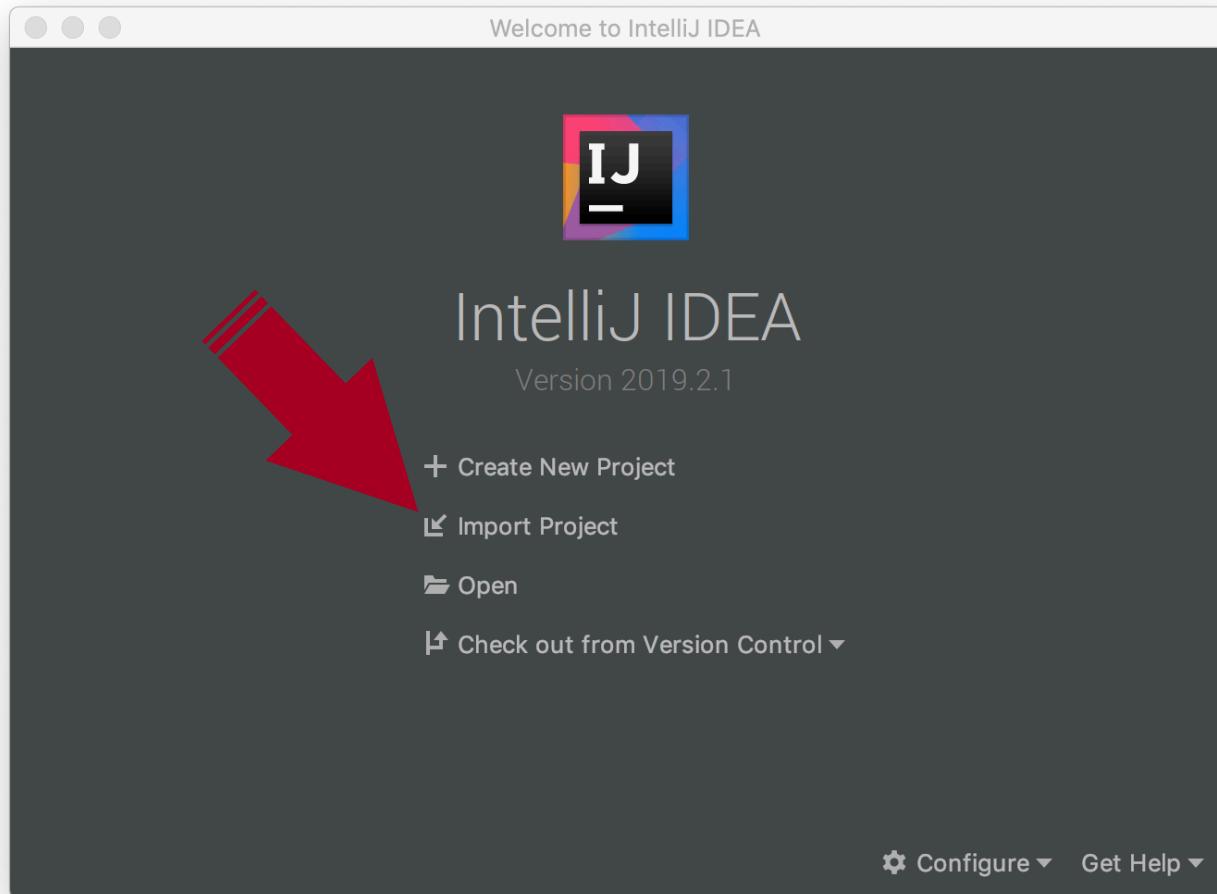
# Download Project

- Download the project
  - [https://drive.google.com/open?id=1NOSsg1alKo\\_ZwLSTTYafw1nCzh4otLa](https://drive.google.com/open?id=1NOSsg1alKo_ZwLSTTYafw1nCzh4otLa)
  - Extract the downloaded file



# Import Project in IntelliJ IDEA

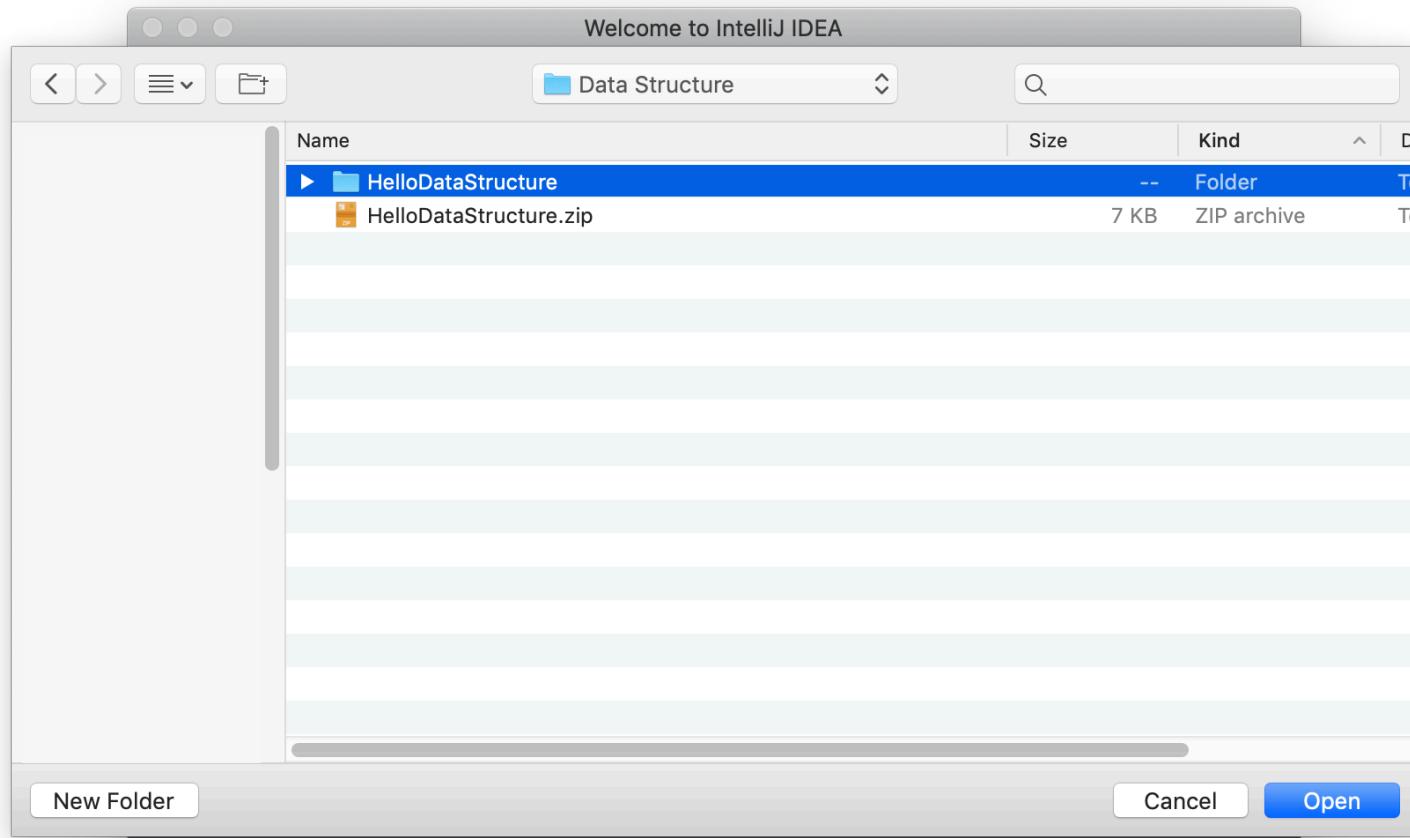
- Click Import Project





# Import Project in IntelliJ IDEA

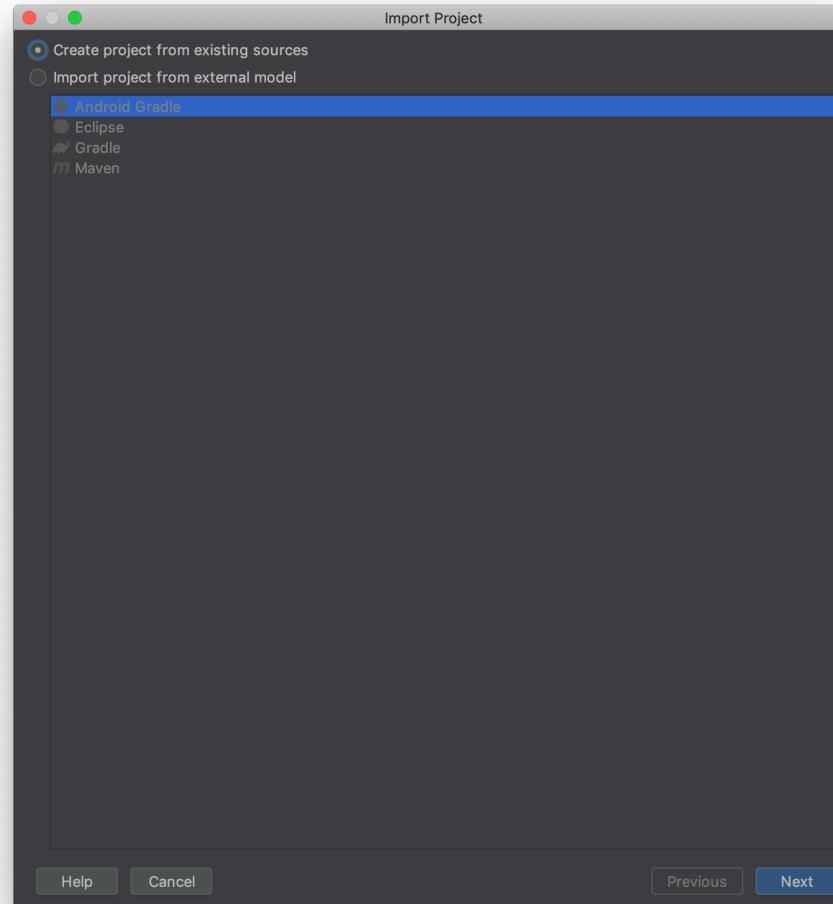
- Navigate to downloaded file and click Open





# Import Project in IntelliJ IDEA

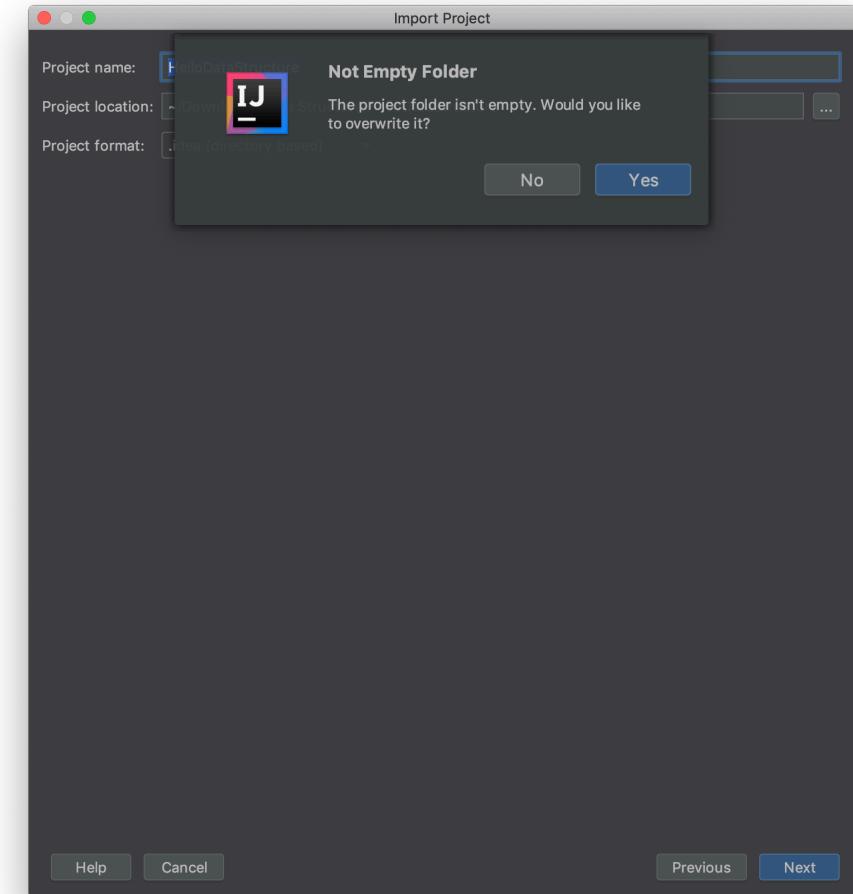
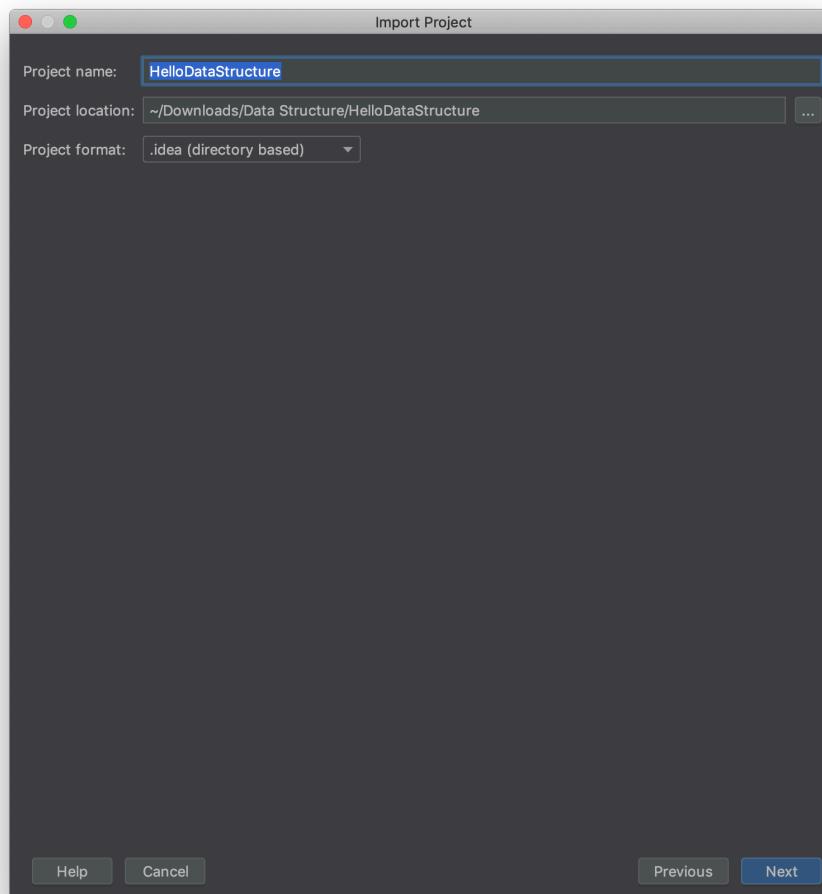
- Click Next





# Import Project in IntelliJ IDEA

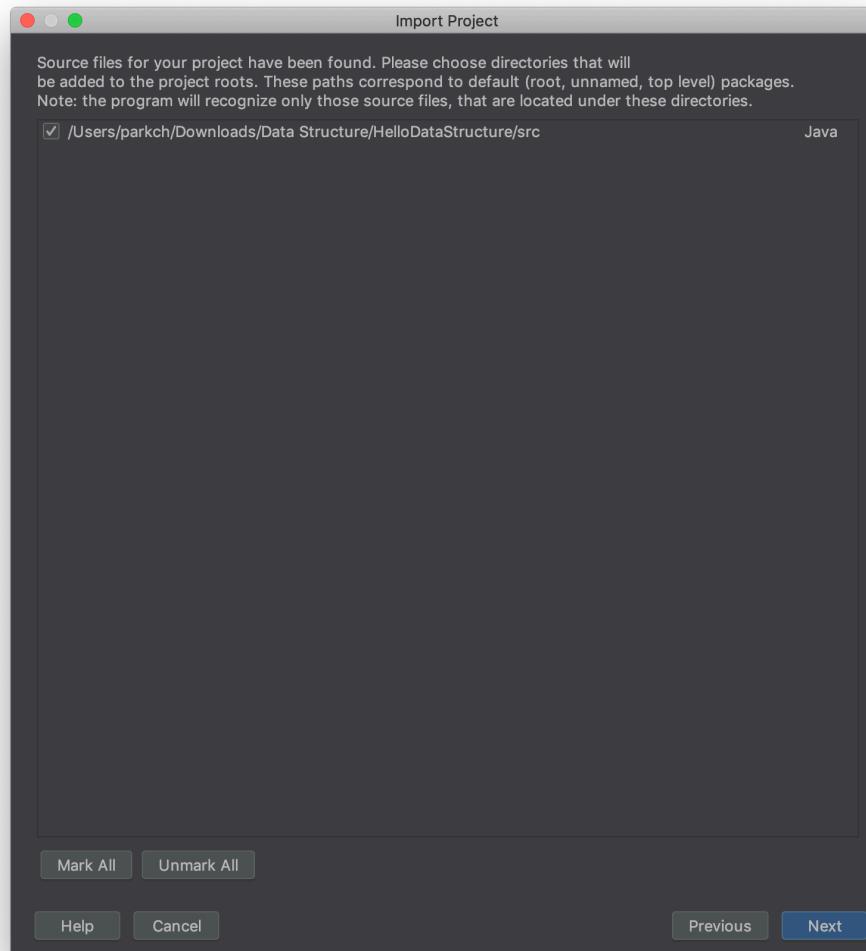
- Click Next and click Yes





# Import Project in IntelliJ IDEA

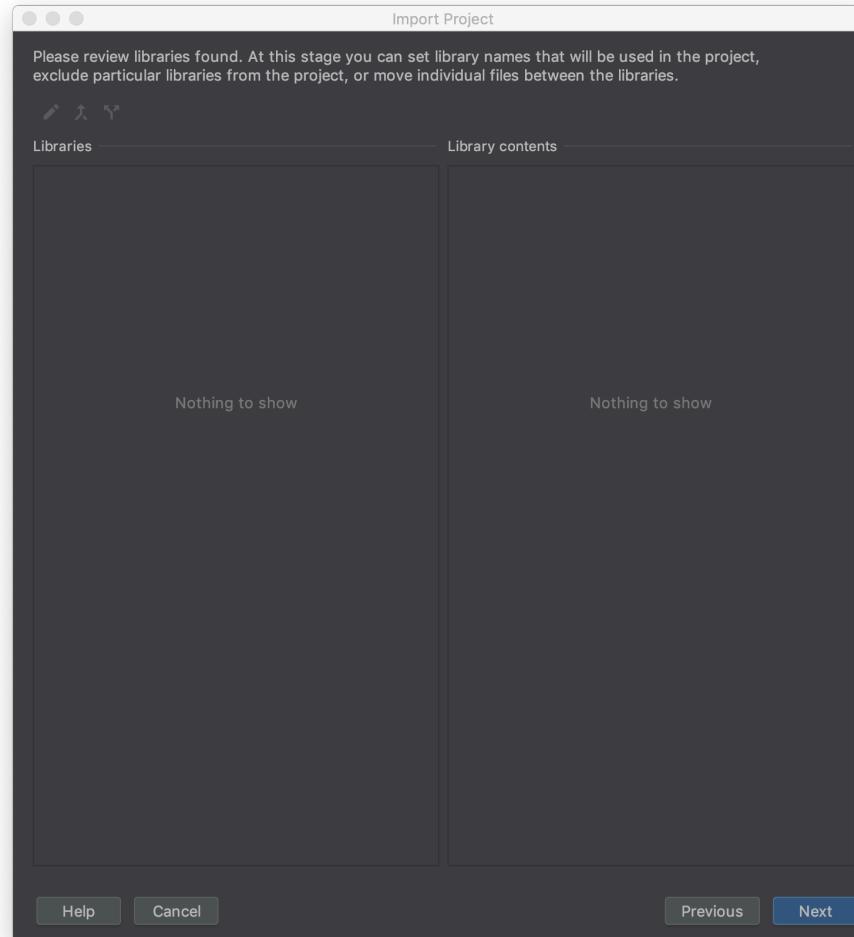
- Click Next





# Import Project in IntelliJ IDEA

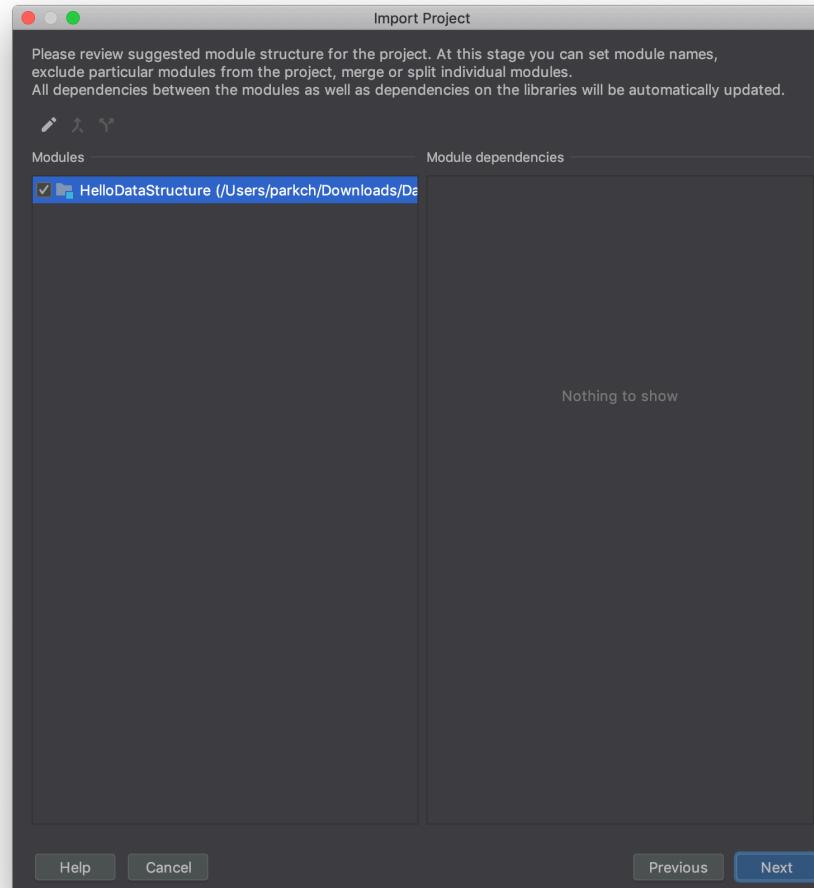
- Click Next





# Import Project in IntelliJ IDEA

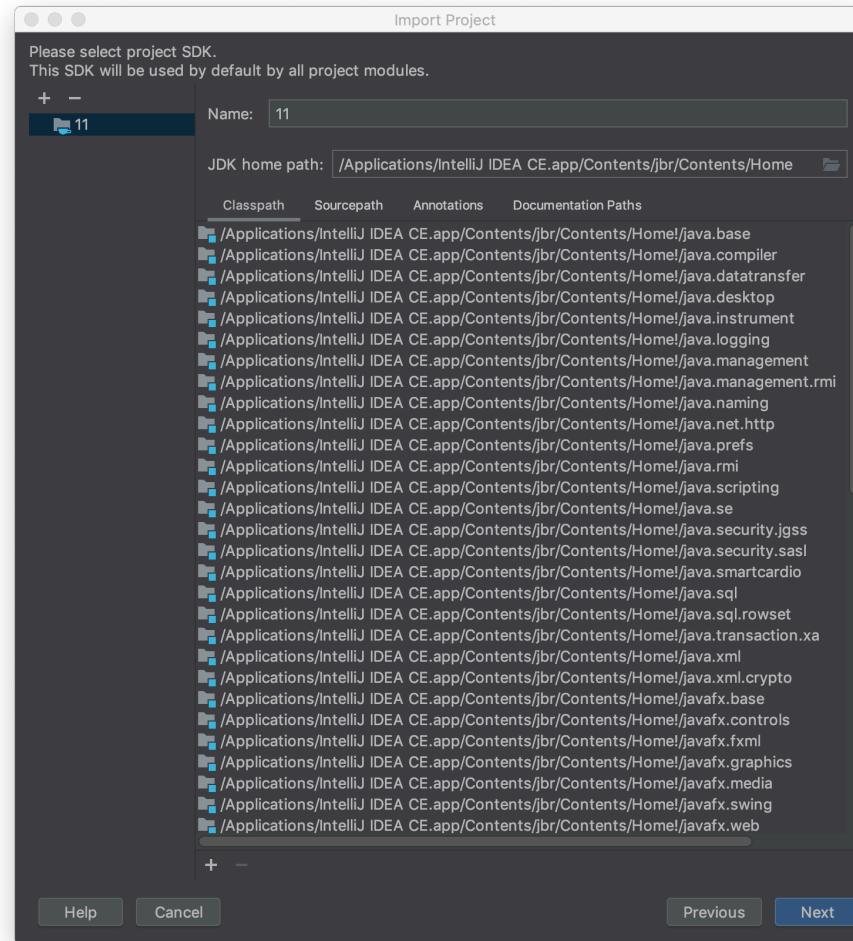
- Click Next





# Import Project in IntelliJ IDEA

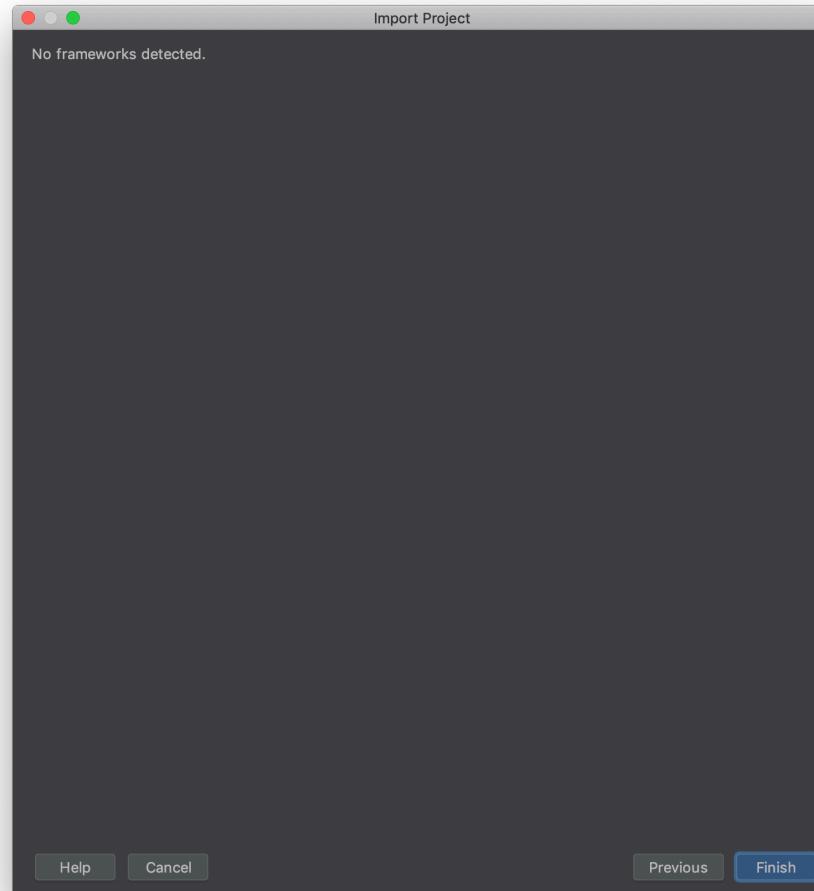
- Click Next





# Import Project in IntelliJ IDEA

- Click Finish





# Run Project in IntelliJ IDEA

- Navigate to Main.java and open
  - Location: ~/src/com.datalab/Main.java

The screenshot shows the IntelliJ IDEA interface with the following details:

- Title Bar:** HelloDataStructure [~/Downloads/Data Structure/HelloDataStructure] - .../src/com/datalab/Main.java
- Project Tool Window:** Shows the project structure: HelloDataStructure (~/Downloads/Data Structure) containing .idea, bin, src, and com.datalab. The com.datalab folder contains a Main.java file.
- Editor:** The Main.java file is open. The code is as follows:

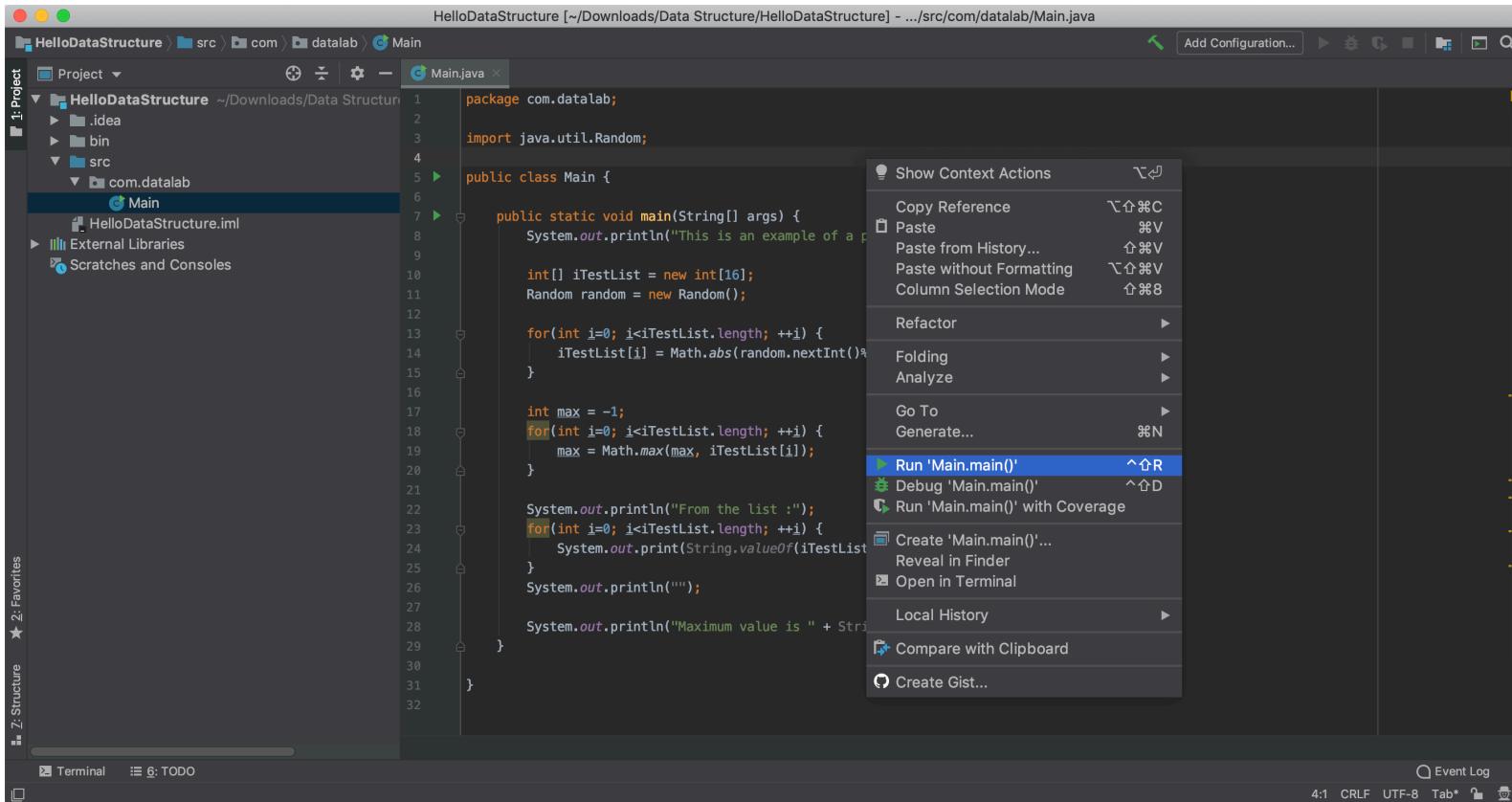
```
package com.datalab;
import java.util.Random;
public class Main {
    public static void main(String[] args) {
        System.out.println("This is an example of a programming assignment");
        int[] iTestList = new int[16];
        Random random = new Random();
        for(int i=0; i<iTestList.length; ++i) {
            iTestList[i] = Math.abs(random.nextInt()%100);
        }
        int max = -1;
        for(int i=0; i<iTestList.length; ++i) {
            max = Math.max(max, iTestList[i]);
        }
        System.out.println("From the list :");
        for(int i=0; i<iTestList.length; ++i) {
            System.out.print(String.valueOf(iTestList[i]) + " ");
        }
        System.out.println("");
        System.out.println("Maximum value is " + String.valueOf(max));
    }
}
```

The code prints a message, generates a list of 16 random integers between 0 and 99, finds the maximum value, and prints the list and maximum value to the console.



# Run Project in IntelliJ IDEA

- Right click on Main.java and click Run Main.main()



The screenshot shows the IntelliJ IDEA interface with the following details:

- Title Bar:** HelloDataStructure [~/Downloads/Data Structure/HelloDataStructure] - .../src/com/datalab/Main.java
- Project Tool Window:** Shows the project structure with packages: HelloDataStructure, com.datalab, and Main.java.
- Main.java Content:** The code implements a class Main with a static main method that prints the maximum value from a randomly generated list of integers.
- Context Menu (Open at Line 17):**
  - Run 'Main.main()' (highlighted)
  - Debug 'Main.main()'
  - Run 'Main.main()' with Coverage
  - Create 'Main.main()...' (disabled)
  - Reveal in Finder
  - Open in Terminal
  - Local History
  - Compare with Clipboard
  - Create Gist...
- Bottom Status Bar:** Event Log, 4:1 CRLF, UTF-8, Tab\*, etc.



# Run Project in IntelliJ IDEA

- You will see the results of the code

The screenshot shows the IntelliJ IDEA interface with the following details:

- Title Bar:** HelloDataStructure [~/Downloads/Data Structure/HelloDataStructure] - .../src/com/datalab/Main.java
- Project Tool Window (Left):** Shows the project structure with a tree view of files and folders. The 'out' folder is selected.
- Main.java Editor (Center):** Displays the Java code:

```
int max = -1;
for(int i=0; i<iTestList.length; ++i) {
    max = Math.max(max, iTestList[i]);
}

System.out.println("From the list :");
for(int i=0; i<iTestList.length; ++i) {
    System.out.print(String.valueOf(iTestList[i]) + " ");
}

System.out.println("");

System.out.println("Maximum value is " + String.valueOf(max));
```
- Run Tool Window (Bottom):** Shows the run configuration 'Main' and the output of the program:

```
/Applications/IntelliJ IDEA CE.app/Contents/jbr/Contents/Home/bin/java" "-javaagent:/Applications/IntelliJ IDEA CE.app/Contents/lib/idea_rt.jar=63927:/Applications/IntelliJ IDEA CE.app
This is an example of a programming assignment
From the list :
51 46 69 93 28 52 49 56 21 32 55 20 29 90 50 72
Maximum value is 93
Process finished with exit code 0
```
- Status Bar (Bottom):** Shows the build status: 'Build completed successfully in 3 s 154 ms (moments ago)' and the current time: '25:10'.



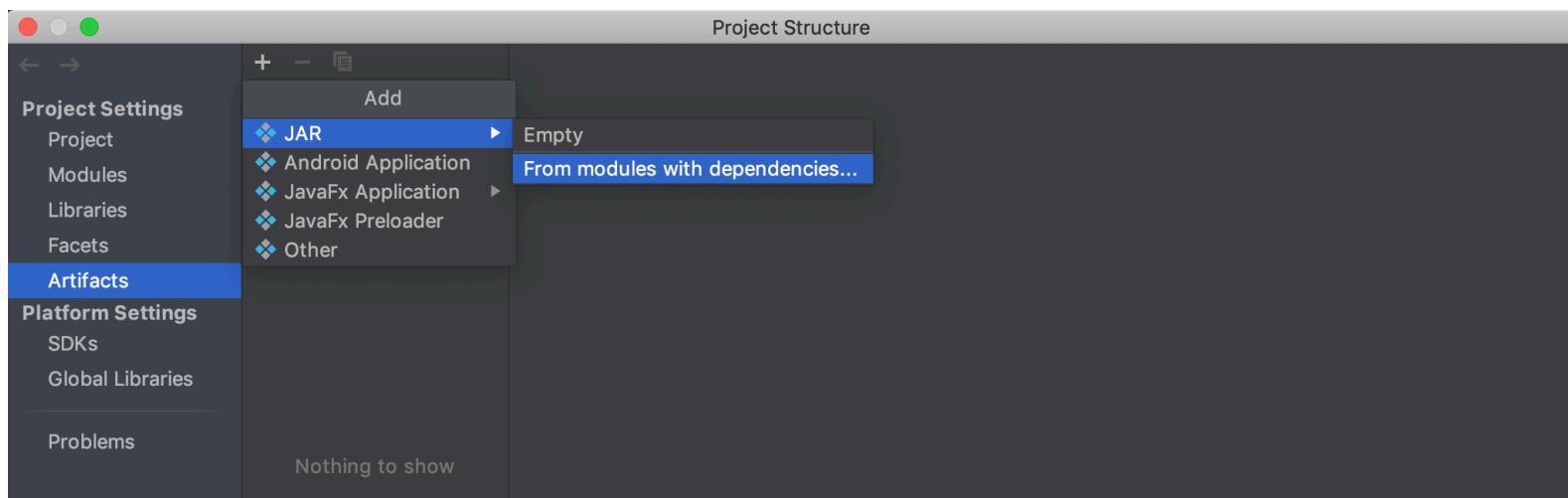
# JAR File

- JAR is an executable file of a Java program
- JAR should be submitted for each assignment
- **You should know how to make a JAR file**



# Create JAR File

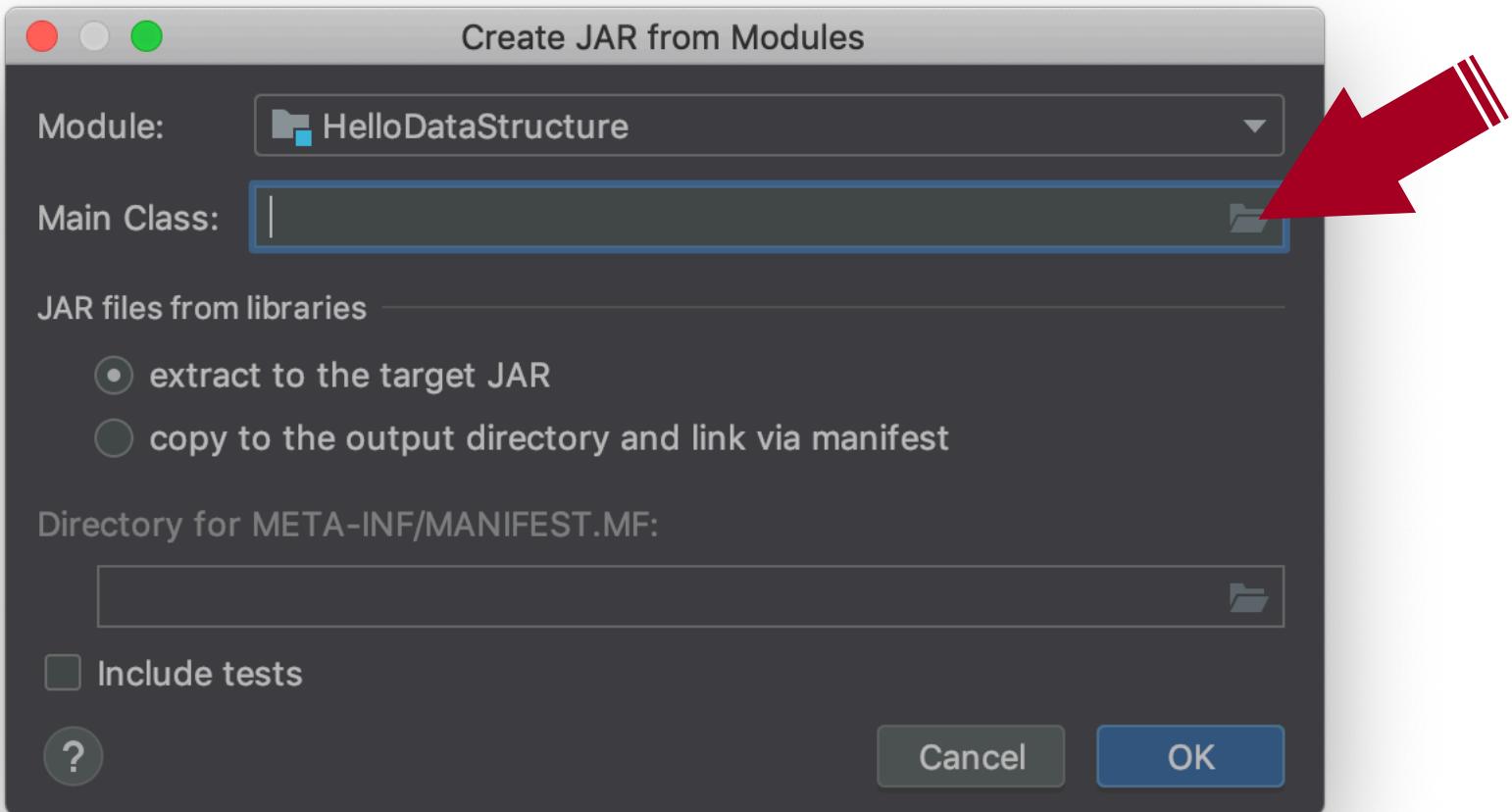
- [File] - [Project Structure...] – [Artifacts]
- Click [+] – [JAR] – [From modules with dependencies...]





# Create JAR File

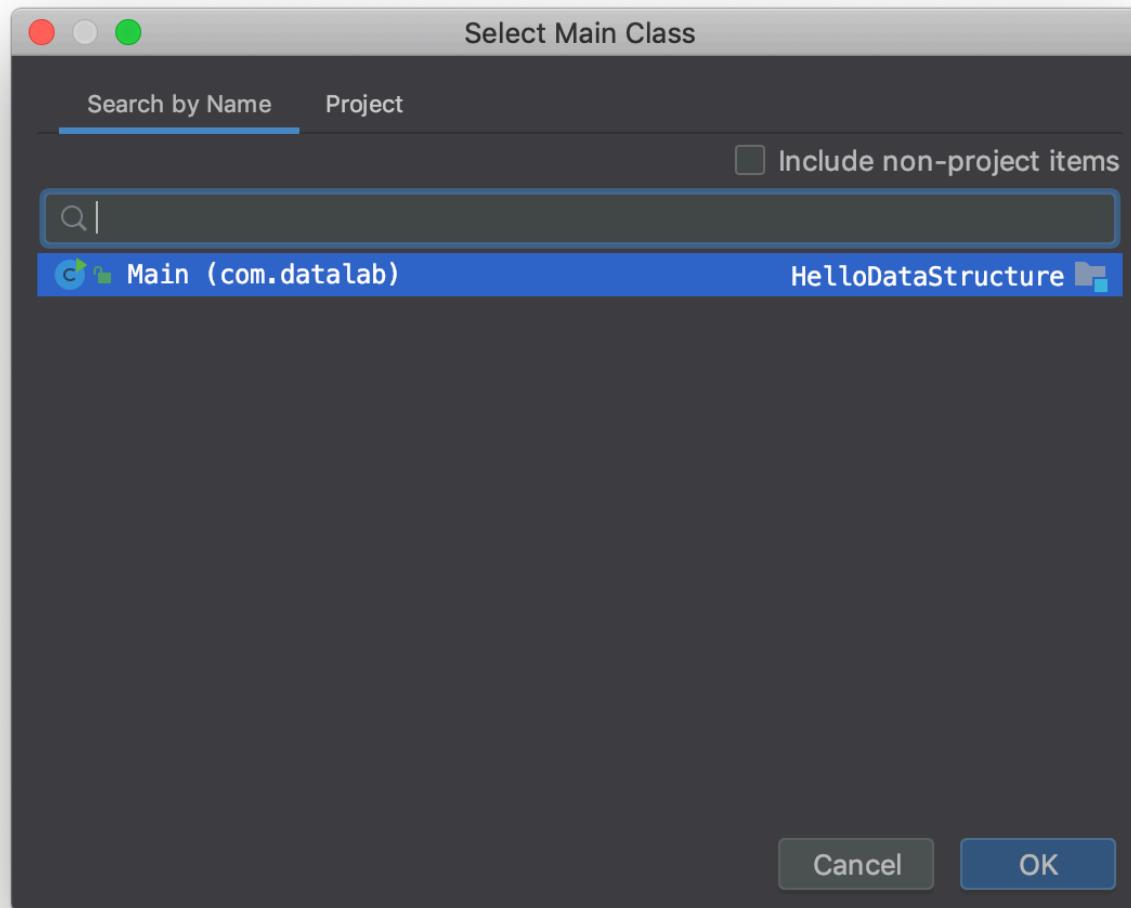
- Click Folder icon in Main Class





# Create JAR file

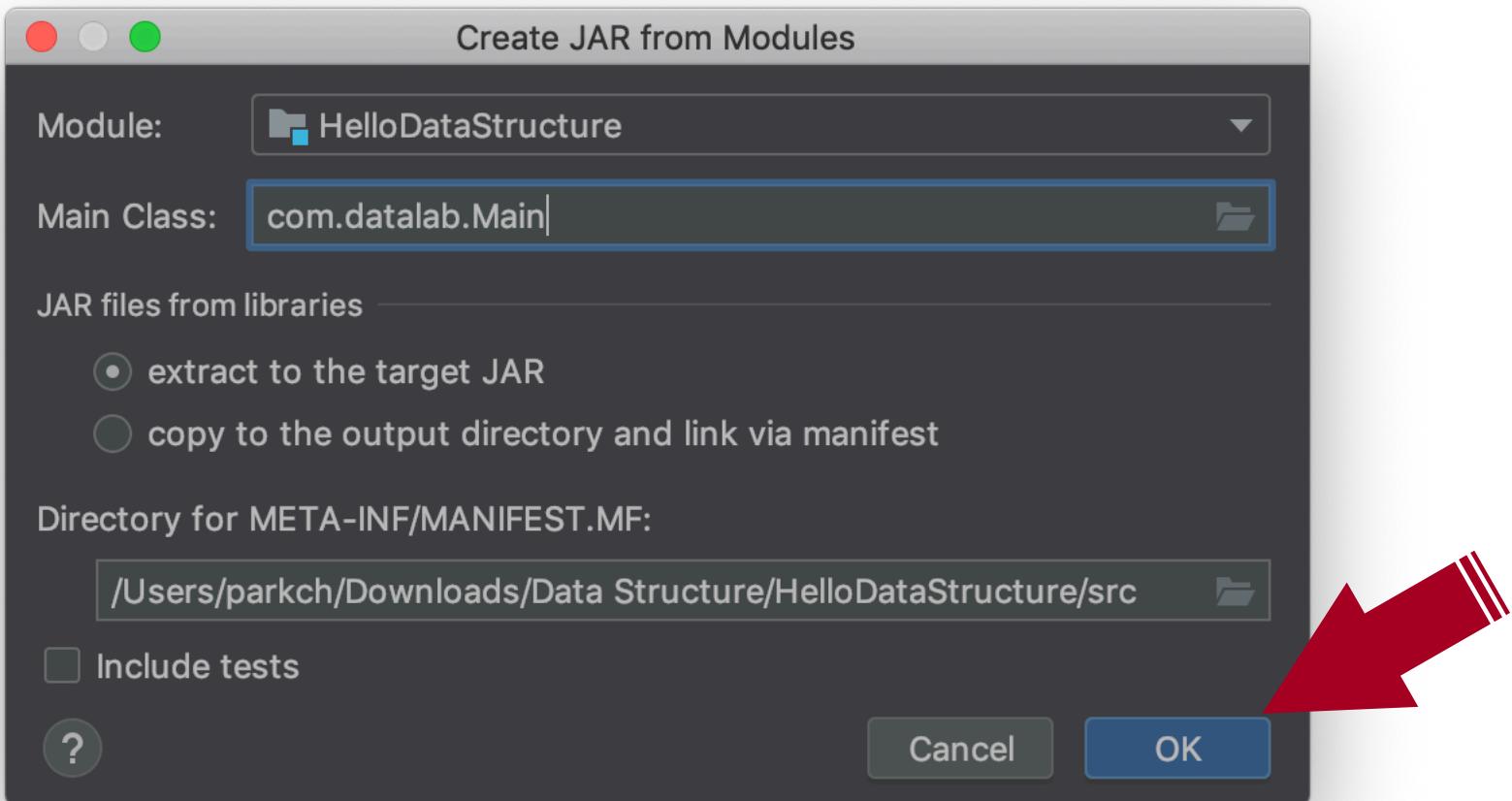
- Click Ok





# Create JAR File

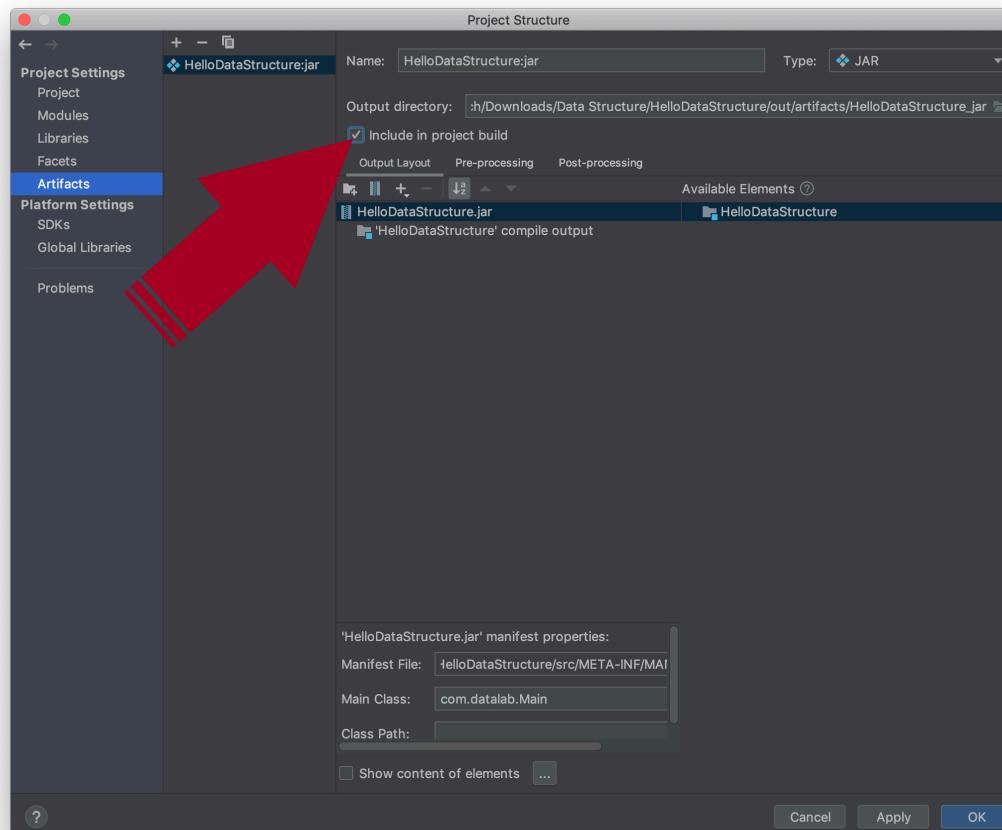
- Click Ok





# Create JAR File

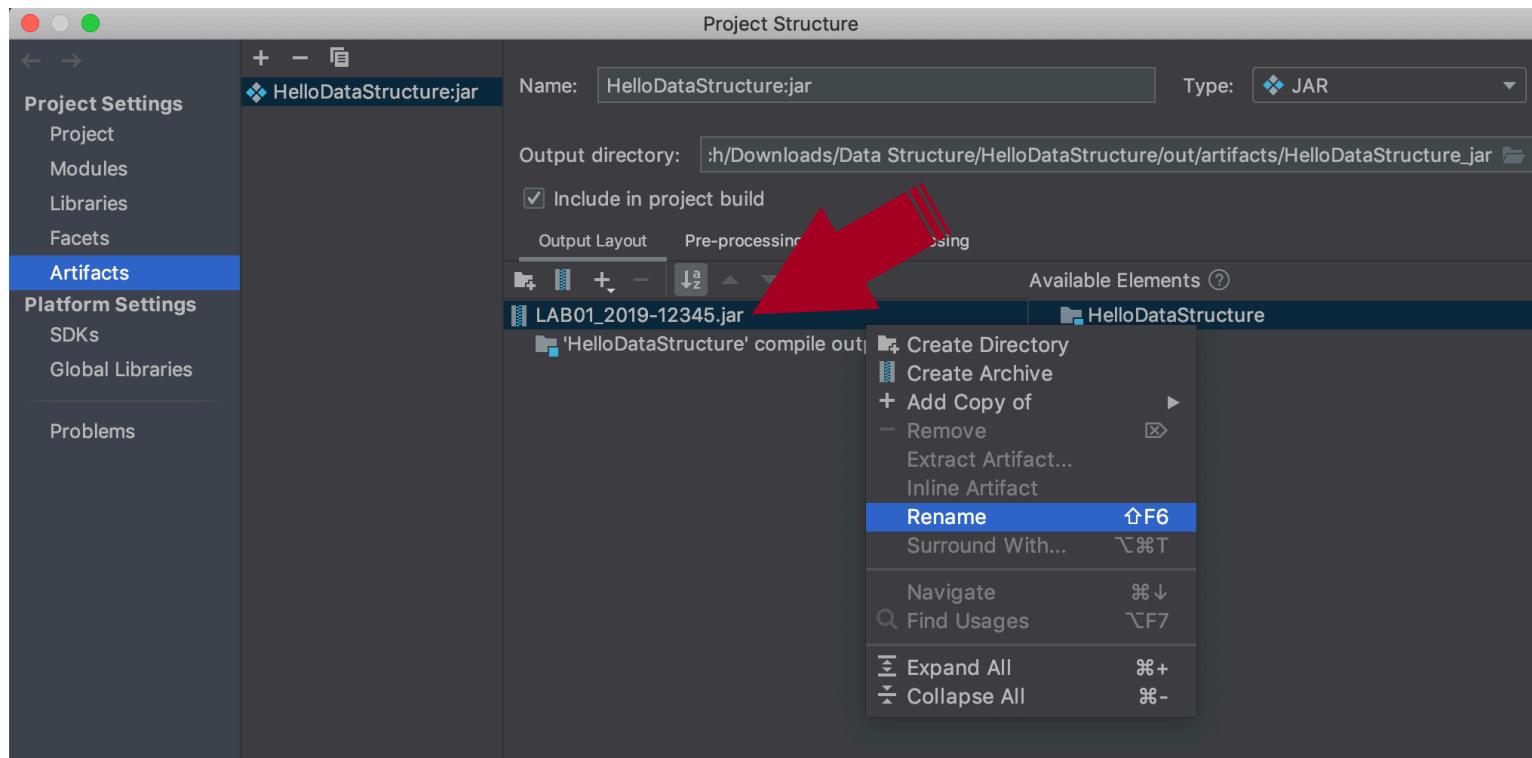
- Click Include in project build
  - The Output directory is where the JAR file is located





# Create JAR File

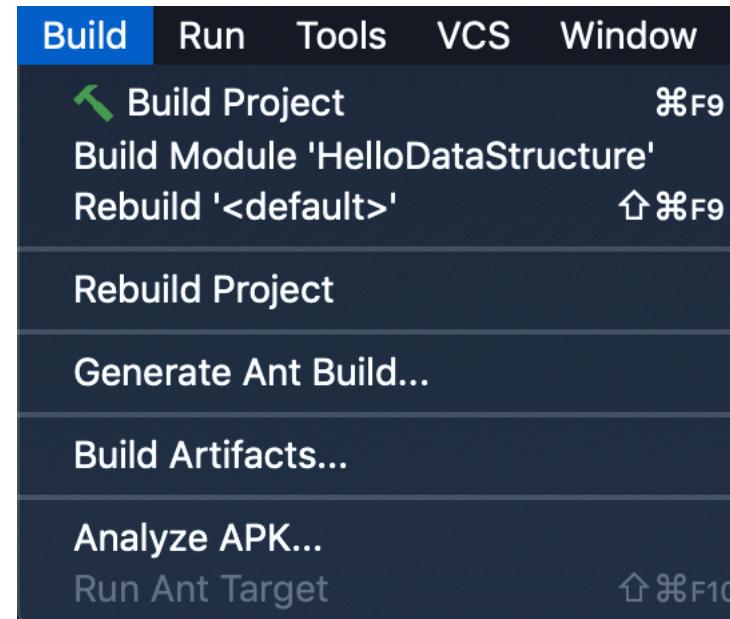
- Right click and rename filename to the following format and click OK when done
  - LAB01\_<Your Student ID>





# Create JAR File

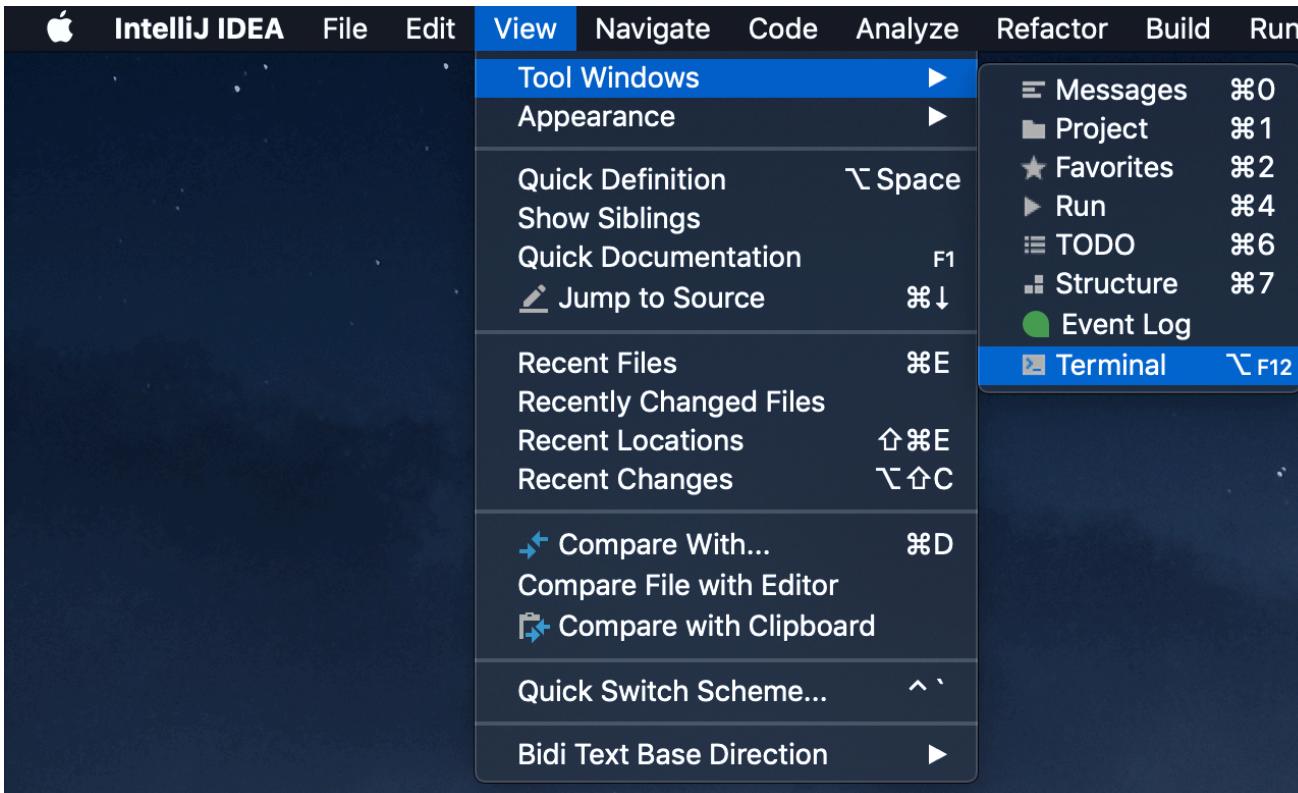
- Build Project
  - [Build] - [Build Project]





# Check your JAR File

- Open Terminal
  - [View] - [Tool Windows] - [Terminal]





# Terminal Commands

Details

## ■ Commands

### □ pwd or chdir (Window)

- Print working directory

```
at 21:16:50 in ~/Downloads
→ pwd
/Users/parkch/Downloads
```

### □ ls or dir (Window)

- List files and directories in current directory

```
at 21:20:04 in Downloads/Data Structure/HelloDataStructure
→ ls
rwxr-xr-x parkch staff 128 B   Tue Sep  3 18:51:00 2019 bin/
rwxr-xr-x parkch staff 128 B   Tue Sep  3 21:02:29 2019 out/
rwxr-xr-x parkch staff 160 B   Tue Sep  3 20:59:45 2019 src/
rw-r--r-- parkch staff 423 B   Tue Sep  3 20:38:31 2019 HelloDataStructure.iml
```



# Terminal Commands

Details

## ■ Commands (cont.)

- `cd <directory name>`
  - Change directory to <directory name>

```
at 21:20:13 in Downloads/Data Structure/HelloDataStructure
→ cd bin

at 21:22:04 in Data Structure/HelloDataStructure/bin
→
```

- `cd ..`
  - Change directory to parent directory

```
at 21:22:04 in Data Structure/HelloDataStructure/bin
→ cd ..

at 21:22:36 in Downloads/Data Structure/HelloDataStructure
→
```



# Check your JAR File

- Navigate to your JAR file using cd command

The screenshot shows a terminal window with the following session:

```
Terminal: Local × +  
at 21:11:03 in Downloads/Data Structure/HelloDataStructure  
→ cd out  
  
at 21:24:45 in Data Structure/HelloDataStructure/out  
→ cd artifacts  
  
at 21:24:47 in HelloDataStructure/out/artifacts  
→ cd HelloDataStructure_jar  
  
at 21:25:12 in out/artifacts/HelloDataStructure_jar  
→  
Build completed successfully in 2 s 647 ms (22 minutes ago)
```

The terminal window has a dark theme. The sidebar on the left shows "Favorites" and "Z: Structure". The bottom status bar includes tabs for Terminal, Messages, Run, TODO, and Event Log, along with build statistics.



# Check your JAR File

## ■ Type Command

- `java -classpath ./<jarFileName.jar> com.datelab.Main`

The screenshot shows a terminal window with the following output:

```
Terminal: Local +  
at 21:47:35 in out/artifacts/HelloDataStructure_jar  
→ java -classpath ./LAB01_2019-12345.jar com.datelab.Main  
This is an example of a programming assignment  
From the list :  
★ 46 61 61 38 43 11 84 12 19 65 94 55 12 99 43 22  
Maximum value is 99  
at 21:47:36 in out/artifacts/HelloDataStructure_jar  
→ []
```

The terminal window also displays the following status bar information:

- 2: Favorites
- Z: Structure
- Terminal 0: Messages 4: Run 6: TODO Event Log
- Build completed successfully in 2 s 714 ms (16 minutes ago) 31:2 CRLF UTF-8 Tab\* 📁 🗑



# Course Information

- T.A.

- Office 301-519
  - Huiwen Xu
    - E-mail: [xuhuiwen33@snu.ac.kr](mailto:xuhuiwen33@snu.ac.kr)
    - Office hour: Wed 13:00 – 14:00
  - Junghoon Kim
    - E-mail: [joseph.junghoon.kim@gmail.com](mailto:joseph.junghoon.kim@gmail.com)
    - Office hour: Thu 13:00 – 14:00
  - Seungcheol Park
    - E-mail: [ant6si@snu.ac.kr](mailto:ant6si@snu.ac.kr)
    - Office hour: Thu 14:00 – 15:00
  - Chaehyun Park
    - E-mail: [chaehyun@snu.ac.kr](mailto:chaehyun@snu.ac.kr)
    - Office hour: Tue 14:00 – 15:00



# Thank You