

JAVA IDE Installation

Day 9
19/07/2025

Famous IDE for Java "IntelliJ IDEA"

Device
Macbook Air M1

The collage consists of several screenshots from the IntelliJ IDEA website:

- A main page for IntelliJ IDEA showing the "Community Edition" section.
- A step-by-step guide:
 - Step 1: Go to Official JetBrains Website
 - Step 2: Choose Community Edition
 - Step 3: Choose Apple Silicon Version
- The "IntelliJ IDEA Community Edition" page, which is free and built on open source.
- A download dialog box asking if you want to allow downloads from "jetbrains.com".
- A progress bar in a Mac OS X-style "Downloads" window showing the download of "ideaIC-2025.1.3-aarch64.dmg".

loading IntelliJ

✓ Step 4: Install IntelliJ IDEA

- Once the .dmg file downloads, double-click it.
- A window appears showing IntelliJ IDEA and Applications folder.
- Drag IntelliJ into the Applications folder.

✓ Step 5: Launch IntelliJ

- Open Launchpad or Spotlight (Cmd + Space → type *IntelliJ IDEA*)
- Open the app.

⚠️ On first launch:

- You may get a message like "App is from an unidentified developer":
- Go to: System Settings → Privacy & Security → Allow Anyway

✓ Step 6: Set Up IntelliJ for First Use

- Choose: Do not import settings (if you're a new user).
- Pick a UI theme (Light/Dark).
- Customize plugins if needed.
- IntelliJ will install JetBrains Runtime.

Ready to Code!

Now you can:

- Create Java/Kotlin/Maven projects.
- Use built-in Gradle support.
- Configure SDK (IntelliJ will detect Java SDK or let you download it).

Bonus: Install Java SDK (if not already)

IntelliJ may prompt you to install a JDK.

- Choose Download JDK option.
- Or install manually from:
 - <https://www.oracle.com/java/technologies/javase-jdk17-downloads.html>
 - Or use Homebrew:

```
bash
brew install openjdk
```

Then link it inside IntelliJ:
Preferences → Build, Execution, Deployment → SDKs

```
Terminal Shell Edit View Window Help
kveeresh --zsh - 204x60
...
--> Downloading https://ghcr.io/v2/homebrew/core/sequoia/77.1/manifests/77.1
Already downloaded: /users/kveeresh/Library/Caches/Homebrew/downloads/54ec23adb942550a474244a334da5616d97e1ec6d59ed9b7368535018bed8a--icu4c@77-77.1.bottle_manifest.json
--> Pouring icu4c@77-77.1.arm64_sequoia.bottle.tar.gz
D /opt/homebrew/Cellar/icu4c@77/77.1/ 277 files, 61.3MB
...
--> Downloading https://ghcr.io/v2/homebrew/core/harfbuzz/11.2.1/manifests/11.2.1
Already downloaded: /users/kveeresh/Library/Caches/Homebrew/downloads/541d81ca7e880678fb64d9f736803fe520fc219f8fb4782fd91fc81f9465aa--harfbuzz-11.2.1.bottle_manifest.json
--> Pouring harfbuzz@11.2.1.1.arm64_sequoia.bottle.tar.gz
D /opt/homebrew/Cellar/harfbuzz/11.2.1.1/ 77 files, 10.2MB
...
--> Installing openjdk dependency: xz
...
--> Downloading https://ghcr.io/v2/homebrew/core/xzstd/manifests/1.5.7
Already downloaded: /users/kveeresh/Library/Caches/Homebrew/downloads/86a115cc1d43ff8a480fd907f812e70a403e1675d8a7223f61bbb08cb2ad27--xz-5.8.1.bottle_manifest.json
--> Pouring xz-5.8.1.arm64_sequoia.bottle.tar.gz
D /opt/homebrew/Cellar/xz-5.8.1/ 32 files, 2.2MB
...
--> Downloading https://ghcr.io/v2/homebrew/core/little-cms2/manifests/2.17
Already downloaded: /users/kveeresh/Library/Caches/Homebrew/downloads/8e5eac9e1df07a0e01662a5a7b0344b31345897c284172f14874c9d329cb85--little-cms2-2.17.bottle_manifest.json
--> Pouring little-cms2@2.17.1.arm64_sequoia.bottle.tar.gz
D /opt/homebrew/Cellar/little-cms2/2.17/ 21 files, 1.4MB
...
--> Installing openjdk
--> Pouring openjdk@24-8.0.3.arm64_sequoia.bottle.tar.gz
D /opt/homebrew/Cellar/openjdk/24-8.0.3/ 32 files, 1.4MB
...
For the system Java wrappers to find this JDK, symlink it with
sudo ln -sfn /opt/homebrew/.../openjdk/jdk /Library/Java/JavaVirtualMachines/openjdk.jdk

openjdk is keg-only, which means it was not symlinked into /opt/homebrew,
because macOS provides similar software and installing this software in
parallel can cause all kinds of trouble.

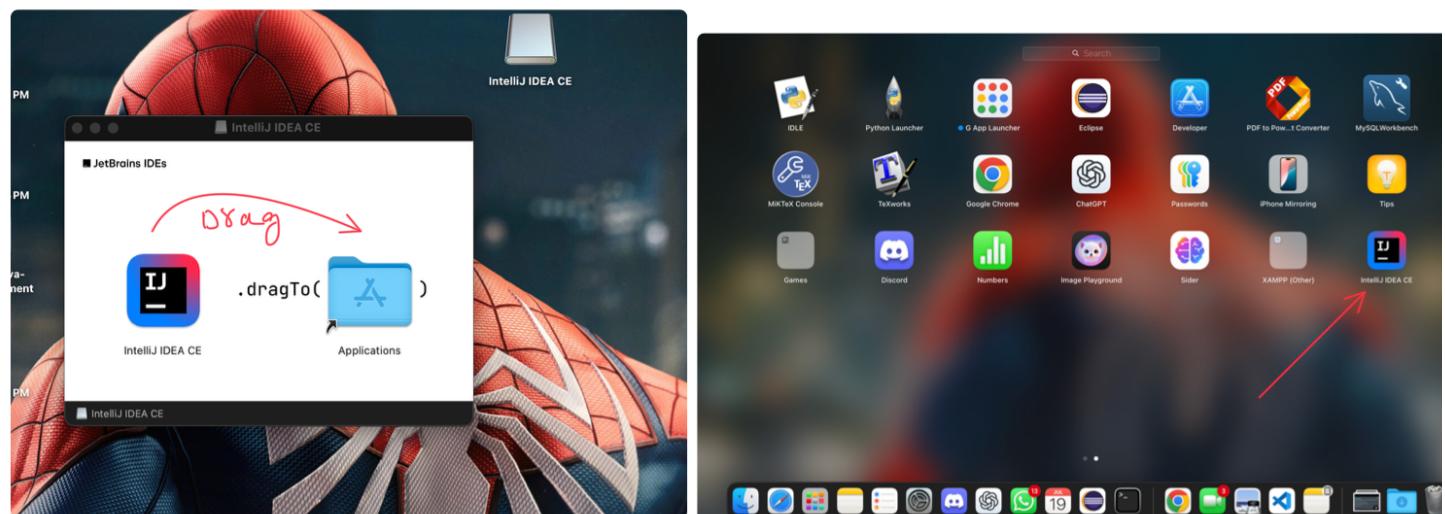
If you need to have openjdk first in your PATH, run:
echo 'export PATH="/opt/homebrew/.../openjdk/bin:$PATH"' >> ~/.zshrc

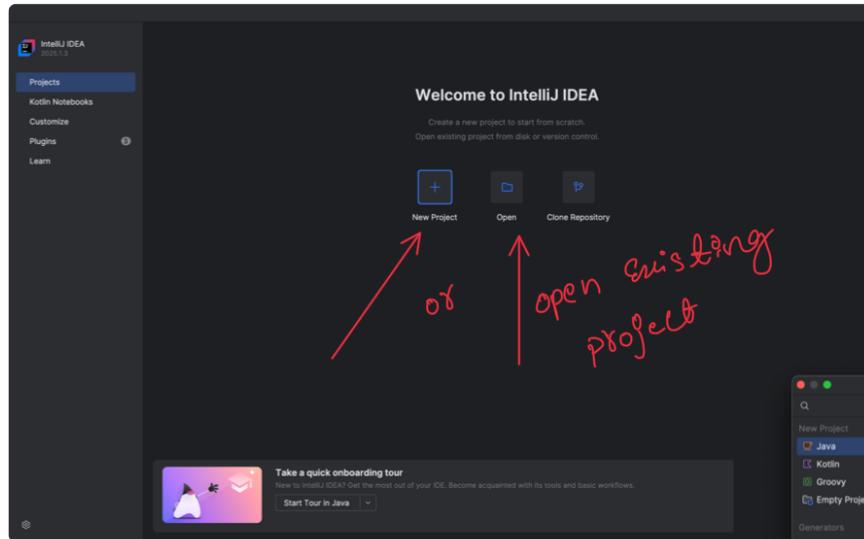
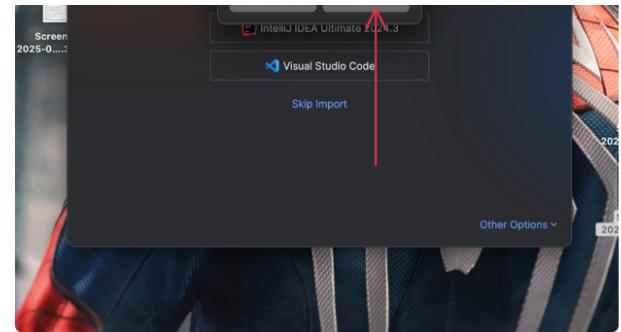
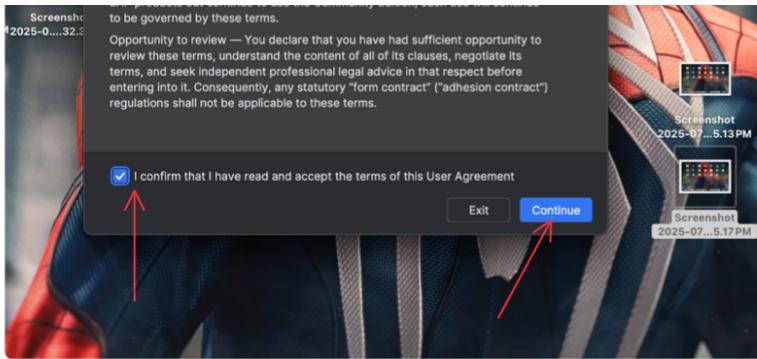
For compilers to find openjdk you may need to set:
export CPPFLAGS="-I/opt/homebrew/.../openjdk/include"
...
--> Summary
D /opt/homebrew/Cellar/openjdk/24-0.1/ 556 files, 368.9MB
Disable this behaviour by setting HOMEBREW_NO_INSTALL_CLEANUP.
Hide these hints with HOMEBREW_NO_ENV_HINTS (see 'man brew').
Removing: /opt/homebrew/Cellar/openjdk/23-0.1... (602 files, 337.2MB)
...
--> openjdk
For the system Java wrappers to find this JDK, symlink it with
sudo ln -sfn /opt/homebrew/.../openjdk/jdk /Library/Java/JavaVirtualMachines/openjdk.jdk

openjdk is keg-only, which means it was not symlinked into /opt/homebrew,
because macOS provides similar software and installing this software in
parallel can cause all kinds of trouble.

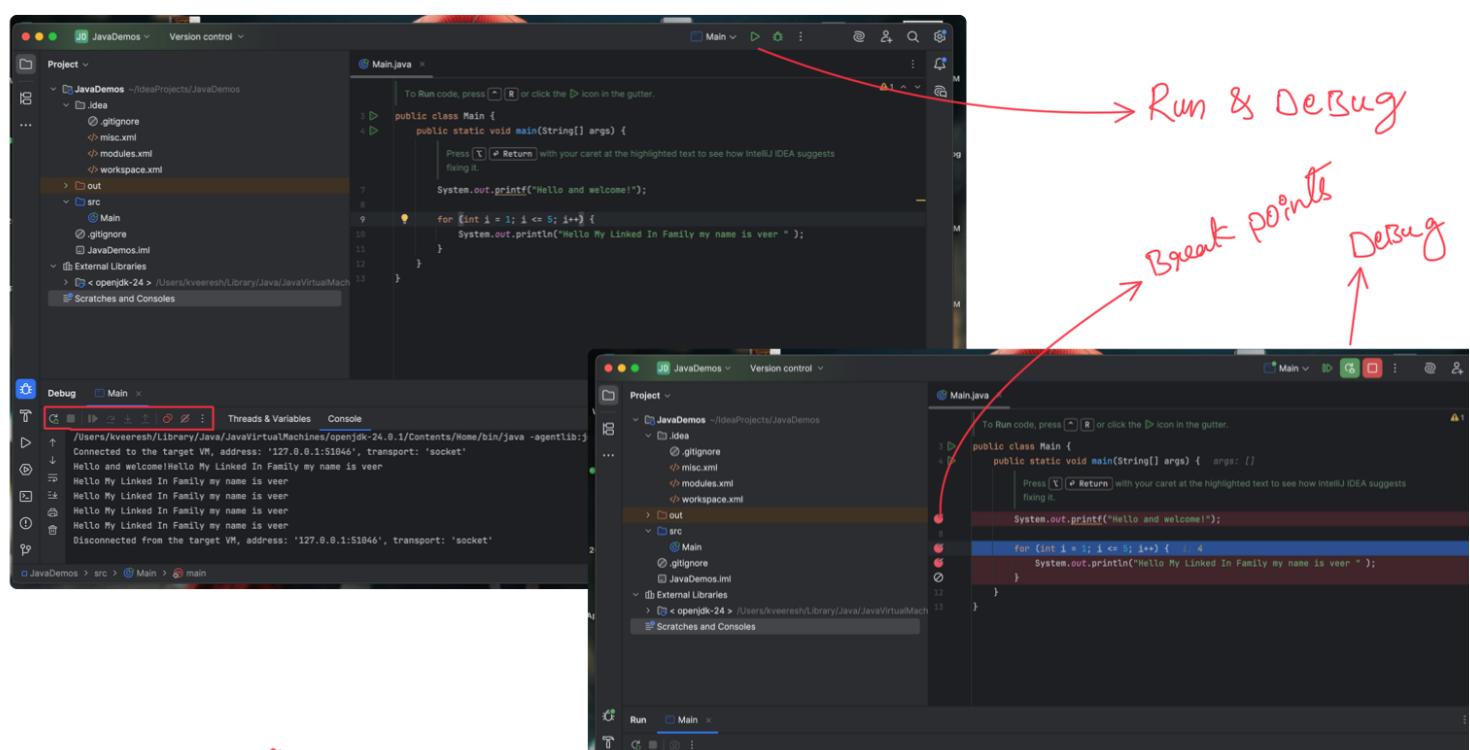
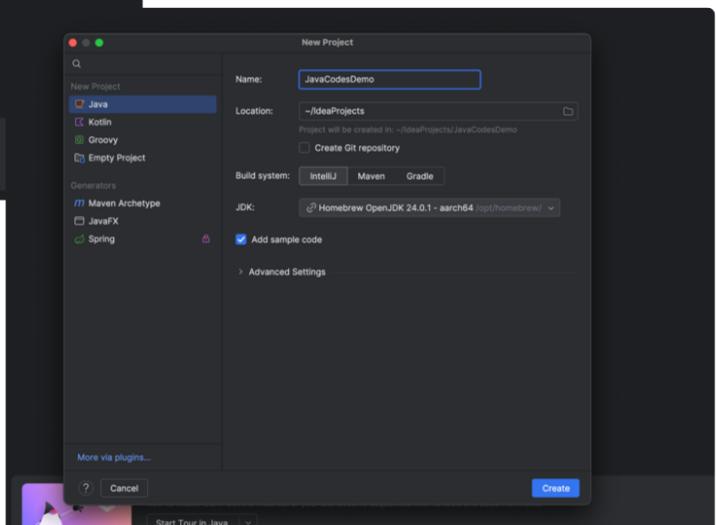
If you need to have openjdk first in your PATH, run:
echo 'export PATH="/opt/homebrew/.../openjdk/bin:$PATH"' >> ~/.zshrc

For compilers to find openjdk you may need to set:
export CPPFLAGS="-I/opt/homebrew/.../openjdk/include"
(base) kveeresh@Ks-MacBook-Air-2 ~ %
```





IDE Installation
Success full



Debugging



```
/Users/kvereresh/Library/Java/JavaVirtualMachines/openjdk-24.0.1/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA CE.app/Contents/lib/idea_rt.jar=
Hello and welcome!
i = 1
i = 2
i = 3
i = 4
i = 5
Process finished with exit code 0
JavaDemos > src > Main
```

Ready To Write Execute De Bug Java code in the "intelliJ IDE"

Comment Down your favourite IDE for writing the code

Java IDE Installation – Steps

Introduction

In this session, we will learn how to set up a **Java development environment**. This includes installing Java (JDK), selecting the right IDE, and writing and running your first Java program.

What is an IDE?

An **IDE (Integrated Development Environment)** is a tool that combines:

- Source code editor
- Compiler/interpreter
- Debugger
- Project manager

It simplifies the process of writing, running, and debugging code.

Popular IDEs for Java

When searching online for “best IDE for Java”, you’ll commonly see these options:

- **IntelliJ IDEA** (by JetBrains) – Most popular and modern IDE
- **Eclipse** – Widely used in the industry, especially for large enterprise apps
- **NetBeans** – Simple and good for beginners
- **BlueJ** – Educational tool for students
- **JDeveloper** – Oracle’s IDE for enterprise Java
- **Android Studio** – For Android app development
- **VS Code** – Lightweight and supports Java via extensions

Use Google or ChatGPT to compare features and choose what suits your needs best.

Step-by-Step Java Development Environment Setup

1. Install Java (JDK)

Java code needs the **Java Development Kit (JDK)** to compile and run.

Options:

- **Oracle JDK** – Official version
- **OpenJDK** – Open-source version

Installation:

1. Visit <https://jdk.java.net> (<https://jdk.java.net>) or <https://www.oracle.com/java> (<https://www.oracle.com/java>)
2. Download for your OS (Windows/Linux/macOS)
3. Install and **add JDK to system environment variables**
 - Set `JAVA_HOME`
 - Add `%JAVA_HOME%/bin` to PATH

Verify Installation:

```
java -version  
javac -version
```

2. Install Your IDE

Option 1: IntelliJ IDEA (Recommended for Java)

1. Go to <https://www.jetbrains.com/idea> (<https://www.jetbrains.com/idea>)
2. Download **Community Edition** (free)
3. Install and launch IntelliJ

Option 2: Eclipse

1. Visit <https://www.eclipse.org> (<https://www.eclipse.org>)
2. Download the Eclipse IDE for Java Developers
3. Install and open Eclipse

Option 3: VS Code

1. Go to <https://code.visualstudio.com> (<https://code.visualstudio.com>)
2. Install **Java Extension Pack**
 - Includes Language Support for Java, Debugger, Maven, etc.

First Java Program

Create a new file called `Main.java` in your IDE or editor of choice.

Code:

```
public class Main {  
    public static void main(String[] args) {  
        System.out.println("Hello My Linked In Family, My name is Veer");  
    }  
}
```

Run It:

- Use the **Run** button in IDE
- Or use terminal:

```
javac Main.java  
java Main
```

Debugging Java

How to Debug:

- Set **breakpoints** by clicking next to the line number.
- Use **Step Over**, **Step Into**, and **Inspect Variables** to trace code execution.

Common Debugging Features:

- Toggle breakpoints
- Watch expressions
- View call stack
- Console output

Think of the debugger like a GPS that helps trace the path your program is taking.

Summary

- Choose an IDE: IntelliJ IDEA, Eclipse, NetBeans, or VS Code.
- Install JDK (Oracle or OpenJDK) and set environment variables.

- Write and run a basic Java program using your chosen IDE.
 - Learn to debug using breakpoints and step-through execution.
 - Use official sources and forums to troubleshoot setup issues.
-