UPES

Object Oriented Programming

Submitted to Dr. Satyam Tiwari

Name: Arnav Sharma
Batch: 2
SAPID: 500120678

Introduction to Java Environment

Java Development Kit (JDK):

This is the toolkit for Java developers. It contains everything needed to *create* Java applications. Think of it as the complete workshop. A crucial part of the JDK is the Java Compiler (javac), which translates human-readable Java code into bytecode (a format the JVM understands). The JDK also includes tools for documentation (javadoc), debugging (jdb), and packaging applications (jar).

Java Runtime Environment (JRE):

This is what you need to *run* Java applications. It's like the finished product showroom. The JRE includes the JVM and all the necessary libraries and resources for executing Java programs. It's enough if you just want to use existing Java software, but it doesn't have the tools for writing your own.

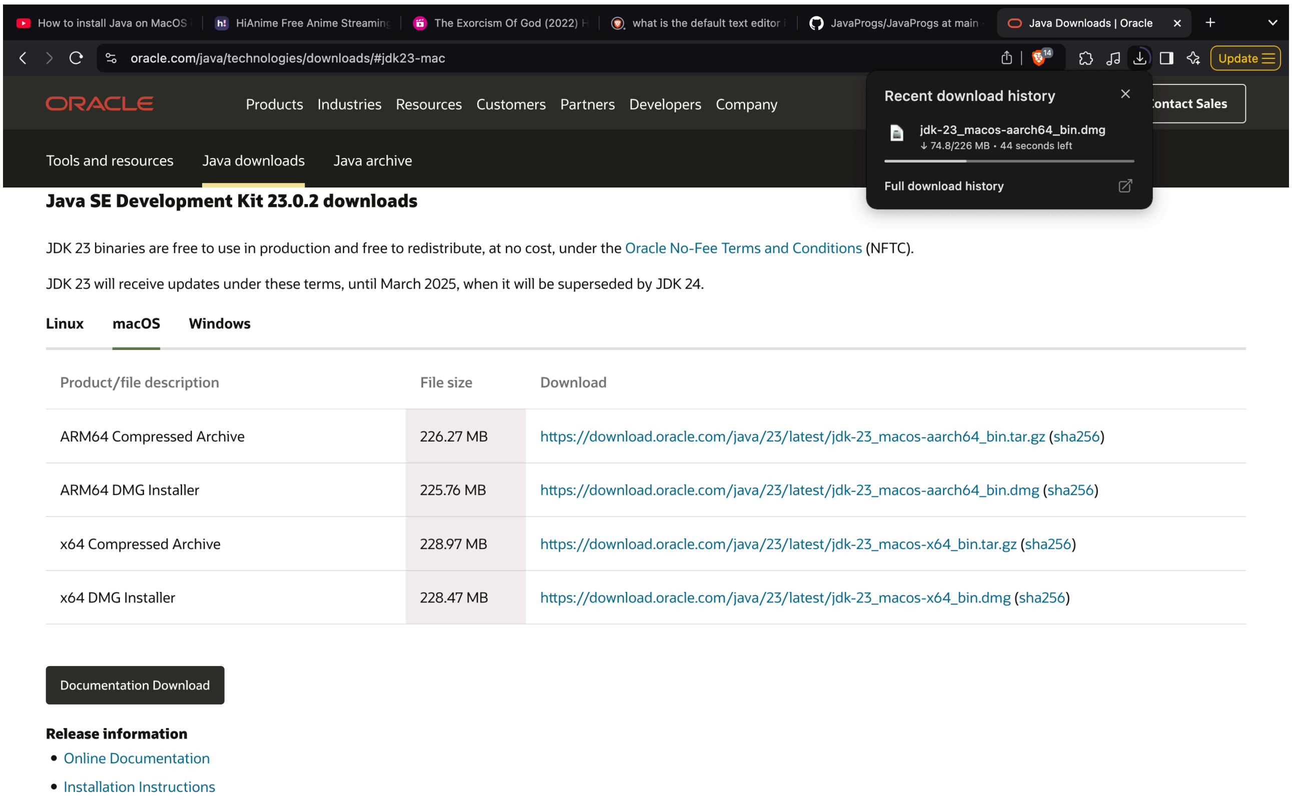
Java Virtual Machine (JVM):

This is the heart of Java's "write once, run anywhere" capability. It's the engine that executes the bytecode. The JVM acts as an intermediary between your Java program and the underlying computer hardware and operating system. It handles tasks like loading and verifying bytecode, converting it to machine code (using Just-In-Time compilation for speed), managing memory (including garbage collection), and dealing with errors. Essentially, the JVM makes sure your Java program runs consistently regardless of the platform it's on.

Conclusion:

In practice, these three components work together seamlessly. Developers use the JDK to write and compile their Java programs. The compilation process converts the source code into byte-code, which is platform-independent and stored in .class files. To run these files, the JRE provides the necessary environment, including the JVM, which interprets or compiles the bytecode into machine code specific to the host system. This layered approach ensures that Java achieves its "Write Once, Run Anywhere" capability, making it one of the most widely used programming languages. The JVM executes the program in a secure and efficient manner, while the JRE ensures that all necessary resources are available, and the JDK provides the tools needed for development. Together, these components form the backbone of the Java ecosystem, supporting developers and end-users alike.

Downloading Latest JDK



How to install Java on MacOS
HiAnime Free Anime Streaming
The Exorcism Of God (2022)
what is the default text editor
JavaProgs/JavaProgs at main
Java Downloads | Oracle
oracle.com/java/technologies/downloads/#jdk23-mac
Update
ORACLE
Products
Industries
Resources
Customers
Partners
Developers
Company
Contact Sales
Recent download history
jdk-23_macos-aarch64_bin.dmg
↓ 74.8/226 MB • 44 seconds left
Full download history
Tools and resources
Java downloads
Java archive

Java SE Development Kit 23.0.2 downloads

JDK 23 binaries are free to use in production and free to redistribute, at no cost, under the Oracle No-Fee Terms and Conditions (NFTC).

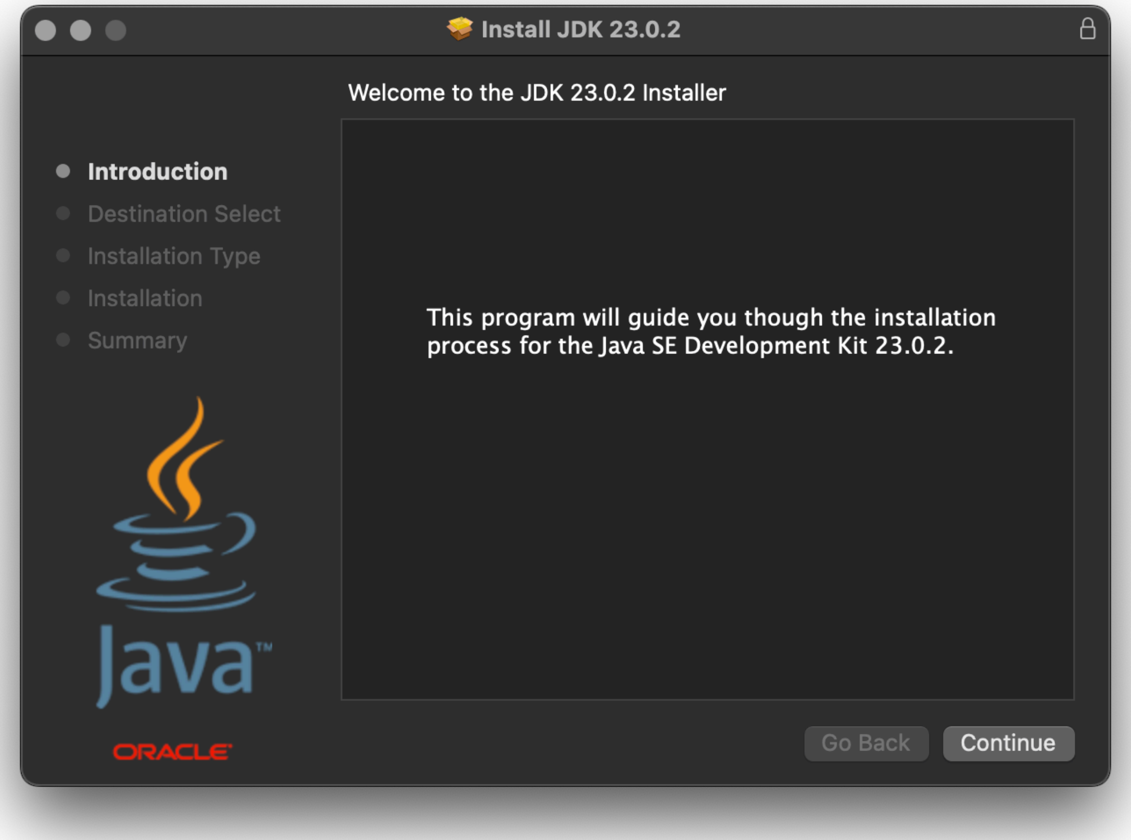
JDK 23 will receive updates under these terms, until March 2025, when it will be superseded by JDK 24.

Product/file description	File size	Download
ARM64 Compressed Archive	226.27 MB	https://download.oracle.com/java/23/latest/jdk-23_macos-aarch64_bin.tar.gz (sha256)
ARM64 DMG Installer	225.76 MB	https://download.oracle.com/java/23/latest/jdk-23_macos-aarch64_bin.dmg (sha256)
x64 Compressed Archive	228.97 MB	https://download.oracle.com/java/23/latest/jdk-23_macos-x64_bin.tar.gz (sha256)
x64 DMG Installer	228.47 MB	https://download.oracle.com/java/23/latest/jdk-23_macos-x64_bin.dmg (sha256)

Release information

- Online Documentation
- Installation Instructions

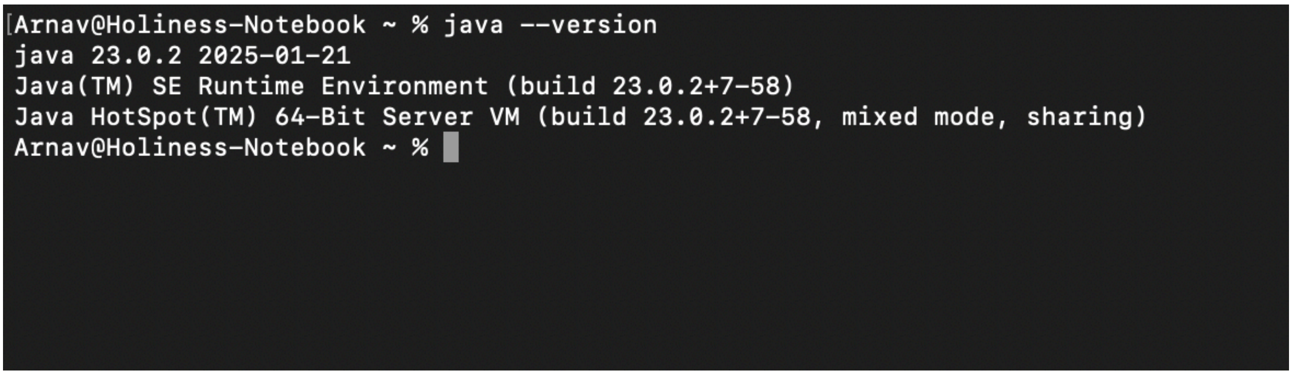
Installing JDK



Welcome to the JDK 23.0.2 Installer

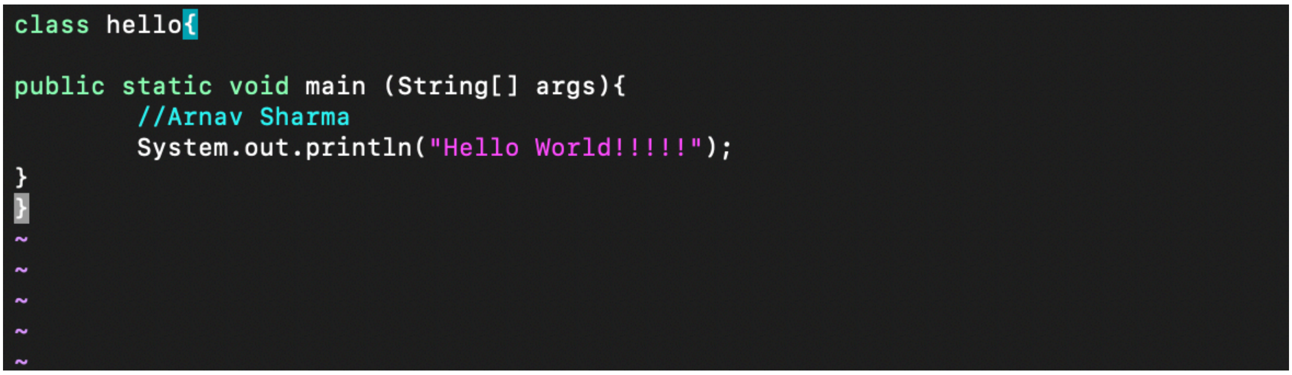
This program will guide you though the installation process for the Java SE Development Kit 23.0.2.

Verifying JDK version



```
[Arnav@Holiness-Notebook ~ % java --version
java 23.0.2 2025-01-21
Java(TM) SE Runtime Environment (build 23.0.2+7-58)
Java HotSpot(TM) 64-Bit Server VM (build 23.0.2+7-58, mixed mode, sharing)
Arnav@Holiness-Notebook ~ % 
```

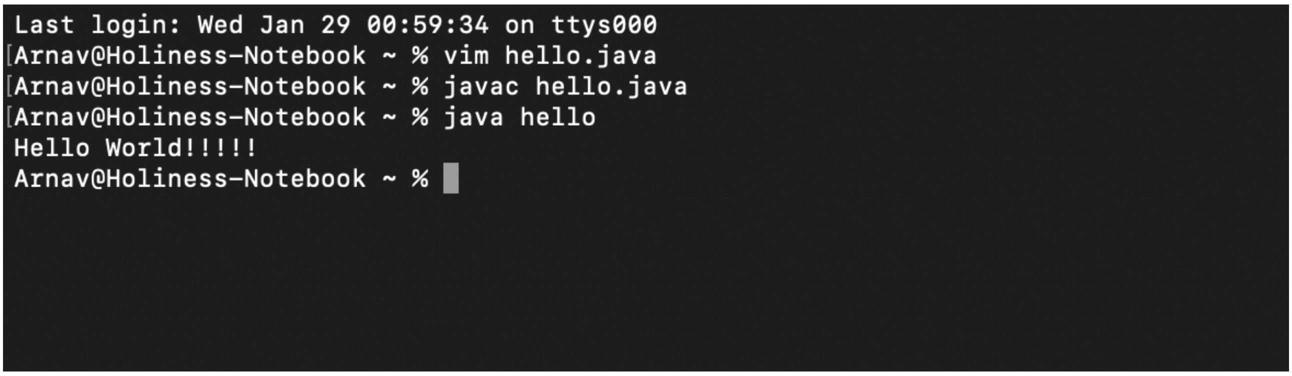
Typing out and saving the Sample Hello World Program in Vim



```
class hello{

public static void main (String[] args){
        //Arnav Sharma
        System.out.println("Hello World!!!!!");
}
}
~
~
~
~
~
```

Compiling, then running the Program



```
Last login: Wed Jan 29 00:59:34 on ttys000
[Arnav@Holiness-Notebook ~ % vim hello.java
[Arnav@Holiness-Notebook ~ % javac hello.java
[Arnav@Holiness-Notebook ~ % java hello
Hello World!!!!!
Arnav@Holiness-Notebook ~ % 
```