

GO :: JDK-18 and JavaFX Installation



JavaFX Installation

JavaFX is an application framework for Java. Java 8 was the final release of Java that included JavaFX in the JDK. In this course we will be using Java 18. With versions of Java after Java 8 you can either install JavaFX separately, or you can use a distribution of the JDK that includes JavaFX. At least for the start of this course, I recommend the latter approach and that's what I will be discussing in this document. Your book discusses manual setup of JavaFX if you're interested.

While you are free to use any IDE, your submissions must not depend on the details of the IDE or any external build process, e.g., Maven. While we may discuss Maven and more complex building processes later in this course, they are unnecessary for the first assignments and if you choose to ignore the advice in this document then I cannot guarantee that I can assist you with debugging any build issues.

I recommend that you download and use IntelliJ. However, it's important that you do not use the built in JavaFX build type. Simply create a standard Java project and use the JavaFX enabled JDK that you will download.

While you are free to use the IDE of your choice, or even no IDE, you must have at least JDK 18 with JavaFX installed.

Java JDK

Before you begin with the IDE installation you will want to make sure that you have an appropriate JDK. If you already have a **Java 18 JDK** installed that is compatible then you may use it and won't need to install a different JDK. I will point out that this is unlikely unless you are an active Java developer and are already doing JavaFX programming. In any case, this is easy to do inside most IDE tools and is not the problem that it used to be with respect to system compatibility. To that end, I will be requiring that you use JDK-18 for this course.

So what does "compatible" mean?

- JDK 18
- Distributed with JavaFX
 - JavaFX is required for this course.

I will not manually dig into your code to see why it doesn't work with a Java 18 JDK!

While there are a number of JDK choices available, please just save yourself the time and effort and download the Azul Zulu OpenJDK. We may install a different JDK later in the course, but for now, this is the shortest path to success.

1. [Azul Zulu Open JDK \(https://www.azul.com/downloads/zulu-community/?architecture=x86-64-bit&package=jdk\)](https://www.azul.com/downloads/zulu-community/?architecture=x86-64-bit&package=jdk)
 - **RECOMMENDED** : This does not have the restrictive license of the standard JDK distribution. Make sure that under *Java Package* that you select JDK-FX.
 - Be sure to download the version of the JDK labeled JDK-FX for Java 18 and your operating system.

While it is possible, and quite easy, to install your JDK anywhere, for most of you I recommend just installing the .msi for Windows, or the .dmg for Mac OS, and allowing the installation to set paths. IntelliJ will find the JDK and this is the shortest path to getting a functioning environment. If you would like, or need, to set this up manually, then we can talk about that in office hours. If enough students have a need to set things up manually then I will create a short instructional video on how to do that.

After you have Java installed and the path pointing to the bin directory, then you should be able to open a command prompt or terminal window and type

```
java -version
```

On my Windows machine it looks something like this. I'm currently using Zulu 11 with Java FX. If you aren't able to get the version to show up, then it's almost certainly just your path. Don't go any further until you get this working.

```
C:\>java -version
openjdk version "18.0.2" 2022-07-19
OpenJDK Runtime Environment Zulu18.32+11-CA (build 18.0.2+9)
OpenJDK 64-Bit Server VM Zulu18.32+11-CA (build 18.0.2+9, mixed mode, sharing)
```

Finally, do not muck with your JAVA_HOME environment variable if it is set for any existing JRE/JDK. You don't need to change that to use the java command from the command line or to use a different JDK inside of an IDE.

Now that you have a JDK working properly, you need to install an IDE of your choice. I use IntelliJ IDEA and I will not be installing either Eclipse or NetBeans to verify any issues that you may be having. You may, of course, ask questions in the discussion forums to see if other students are able to help you out. I will do my best to help you based on my past experience with these other IDEs, but if you have no preference and you are not strong with IT issues, then I highly recommend that you install IntelliJ. All in class and video discussions will use IntelliJ. I may use VSCode, which is a

build of VSCode with Microsoft's telemetry removed, but don't think that you have to use either VSCode or VSCodium, it simple makes some demos easier.

IntelliJ IDEA Installation

Download and install the Community Edition from the following URL:

<https://www.jetbrains.com/idea/download/#section=windows>

(<https://www.jetbrains.com/idea/download/#section=windows>)

As I mentioned, you will only need to create a standard Java project for this. Doing so will reduce some pain in using the IDE.

From within the IDE, you will need to go to file->Project Structure to change the JDK for your project. Go to Project:Project SDK and click on the down arrow to select **+ Add SDK**. Choose JDK and navigate to your installed JDK.