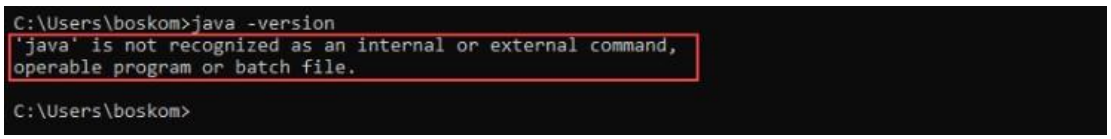
# Installing JDK on Windows

## Check if Java Is Installed

Before installing the Java Development Kit, check if a Java version is already installed on Windows. Follow the steps below:

1. Open a command prompt by typing *cmd* in the search bar and press Enter.

2. Run the following command: **java -version**
   **The command outputs the Java version on your system.**

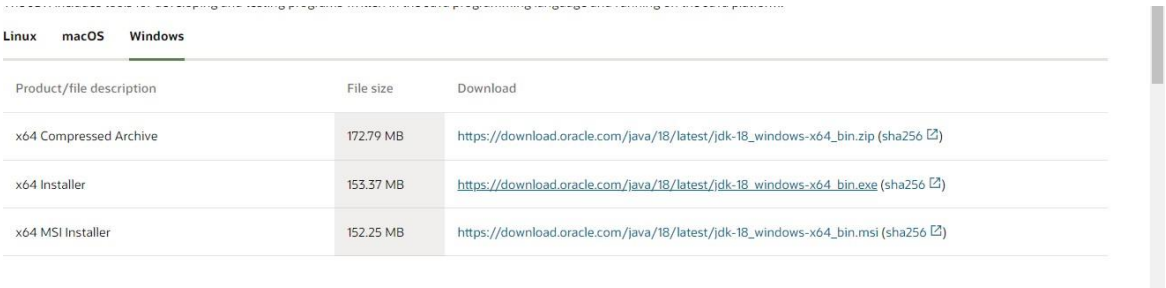



If Java isn't installed, the output is a message stating that Java isn't recognised as an internal or external command.

## Download Java for Windows 10/11

Download the latest Java Development Kit installation file for Windows 10 to have the latest features and bug fixes.

1. **Using your preferred web browser, navigate to the Oracle Java Downloads page.**
2. **On the *Downloads* page, click the x64 Installer download link under the Windows category.**




**Wait for the download to complete.**

# Install Java on Windows 10/11

After downloading the installation file, proceed with installing Java on your Windows system.Follow the steps below:

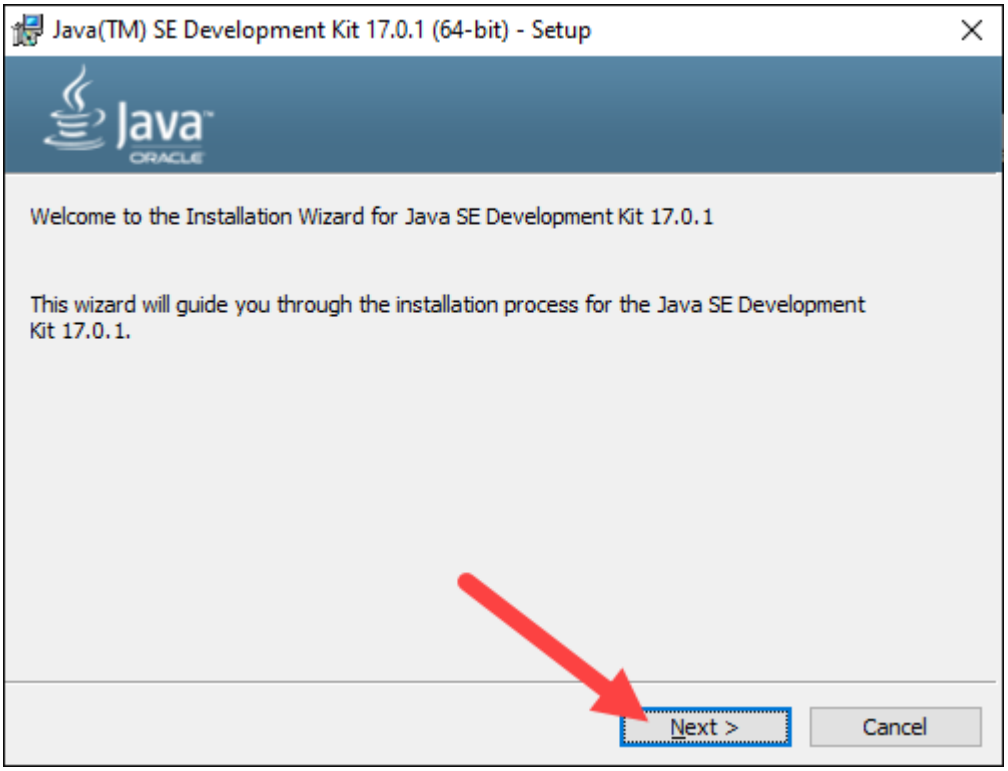
**Step 1: Run the Downloaded File**

Double-click the downloaded file to start the installation.

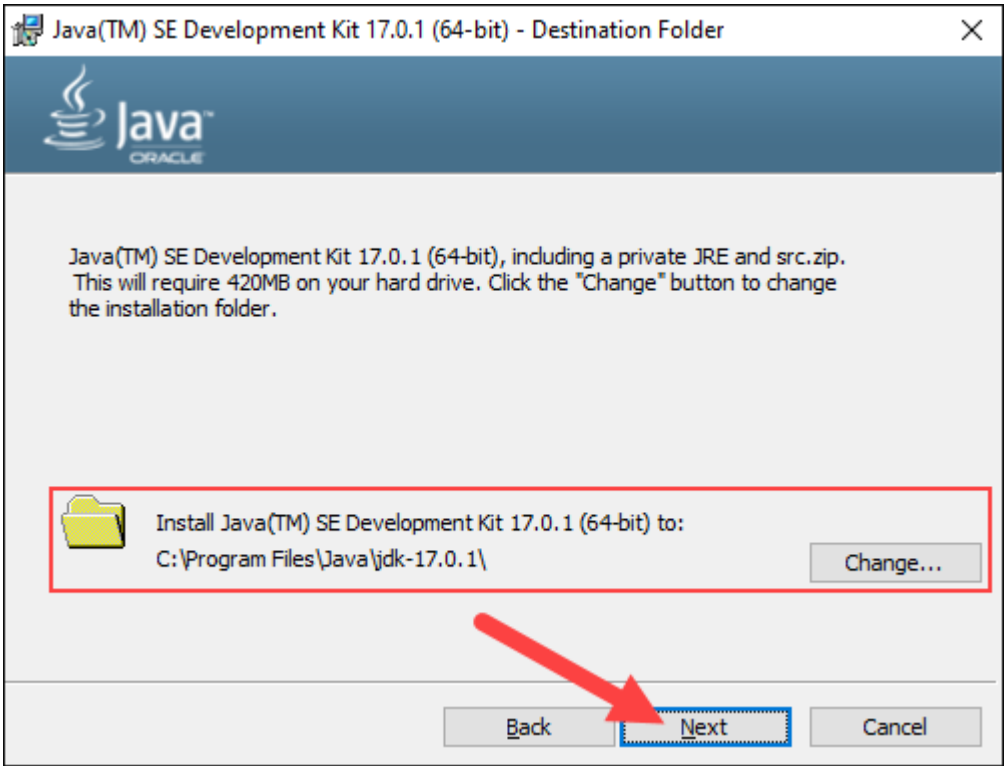
**Step 2: Configure the Installation Wizard**

After running the installation file, the installation wizard welcome screen appears.

**1. Click Next to proceed to the next step.**




**2. Choose the destination folder for the Java installation files or stick to the default path. Click Next to proceed.**




**3. Wait for the wizard to finish the installation process until the *Successfully Installed* message appears. Click Close to exit the wizard.**

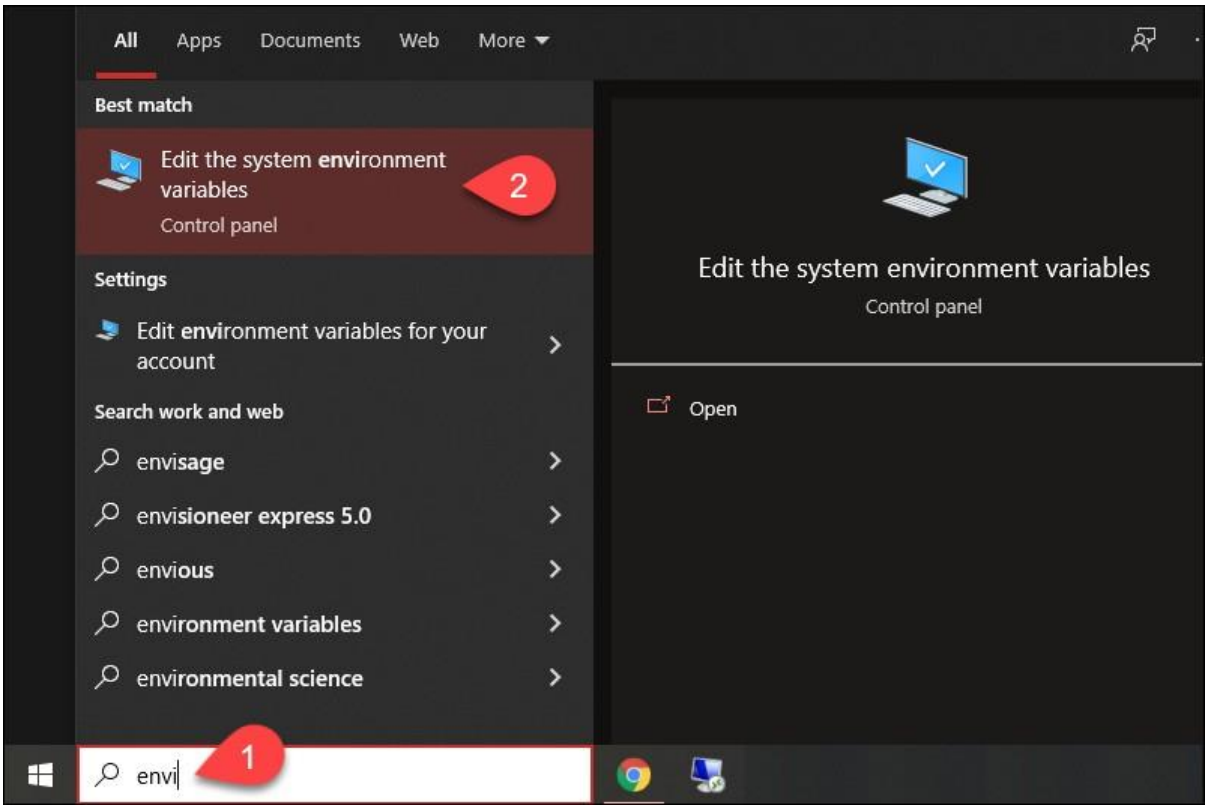

**Set Environment variables in Java**

Set Java environment variables to enable program compiling from any directory. To do so, follow the steps below:

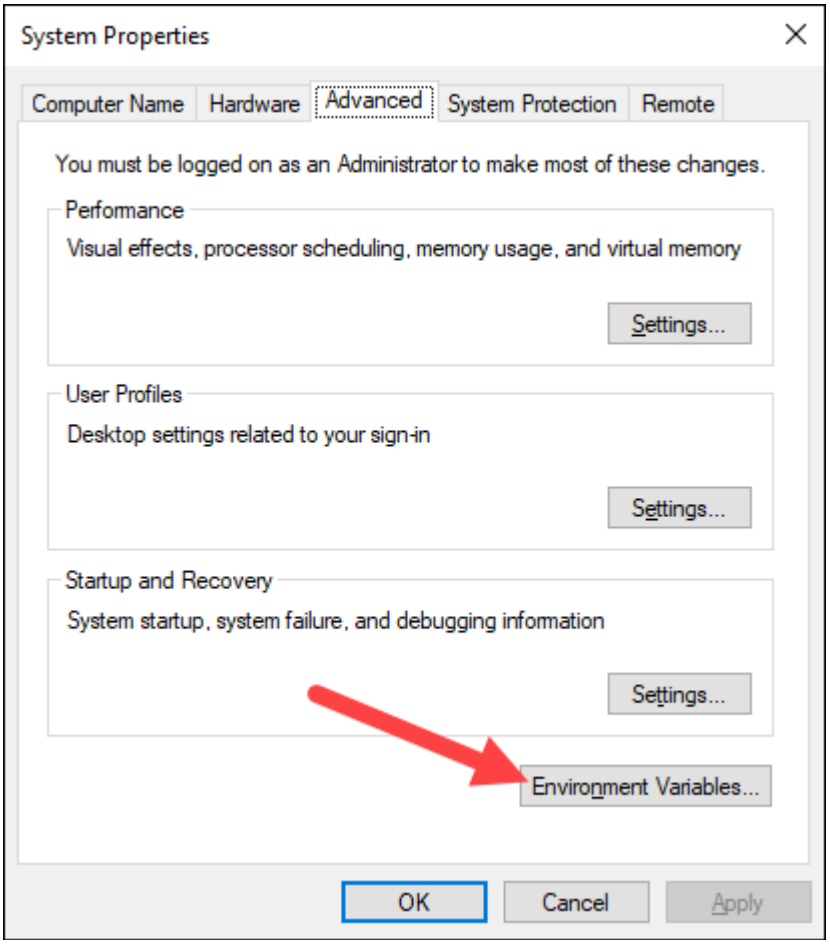
## Step 1: Add Java to System Variables

1. Open the Start menu and search for *environment variables*.

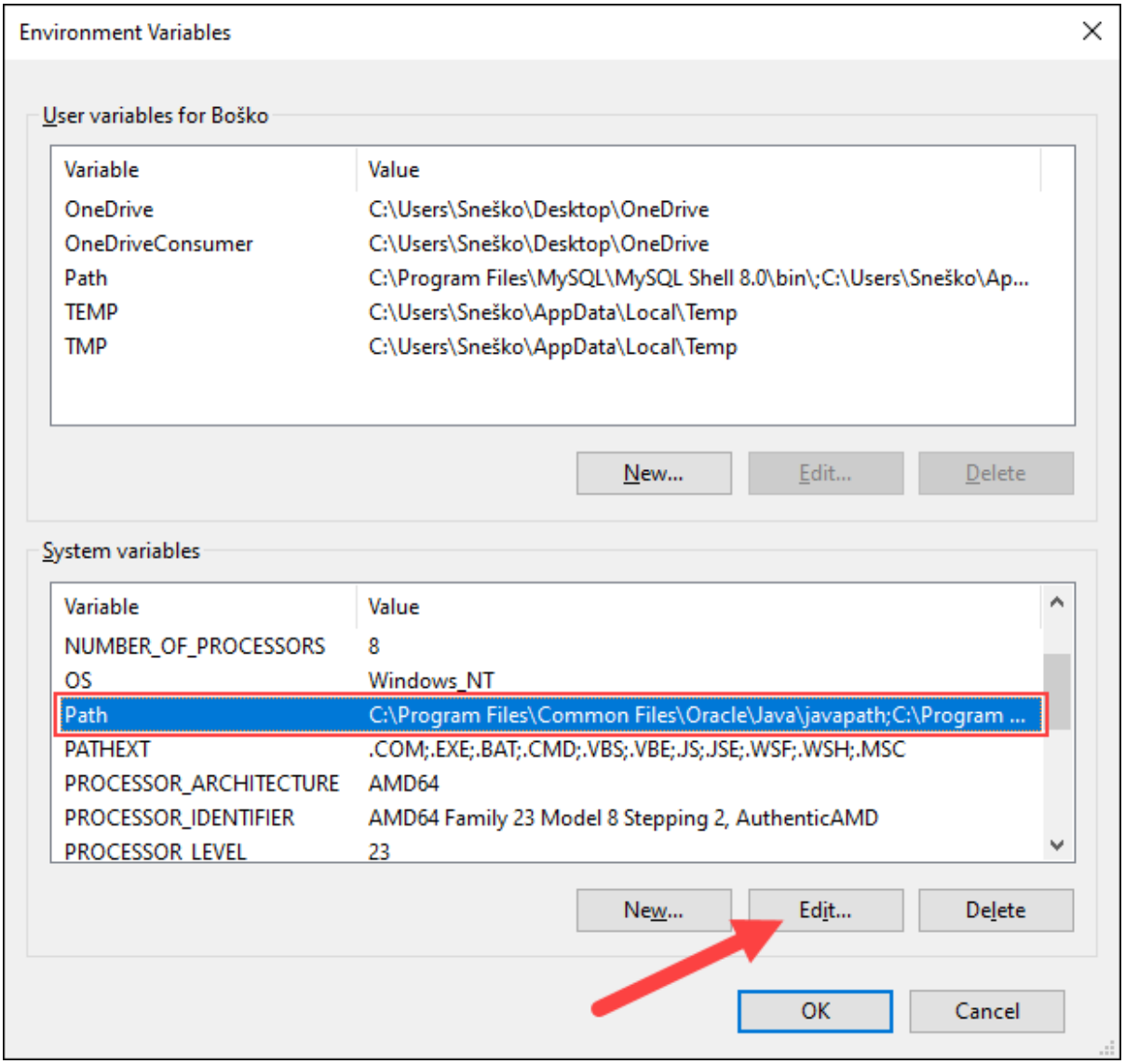
2. Select the Edit the system environment variables result.



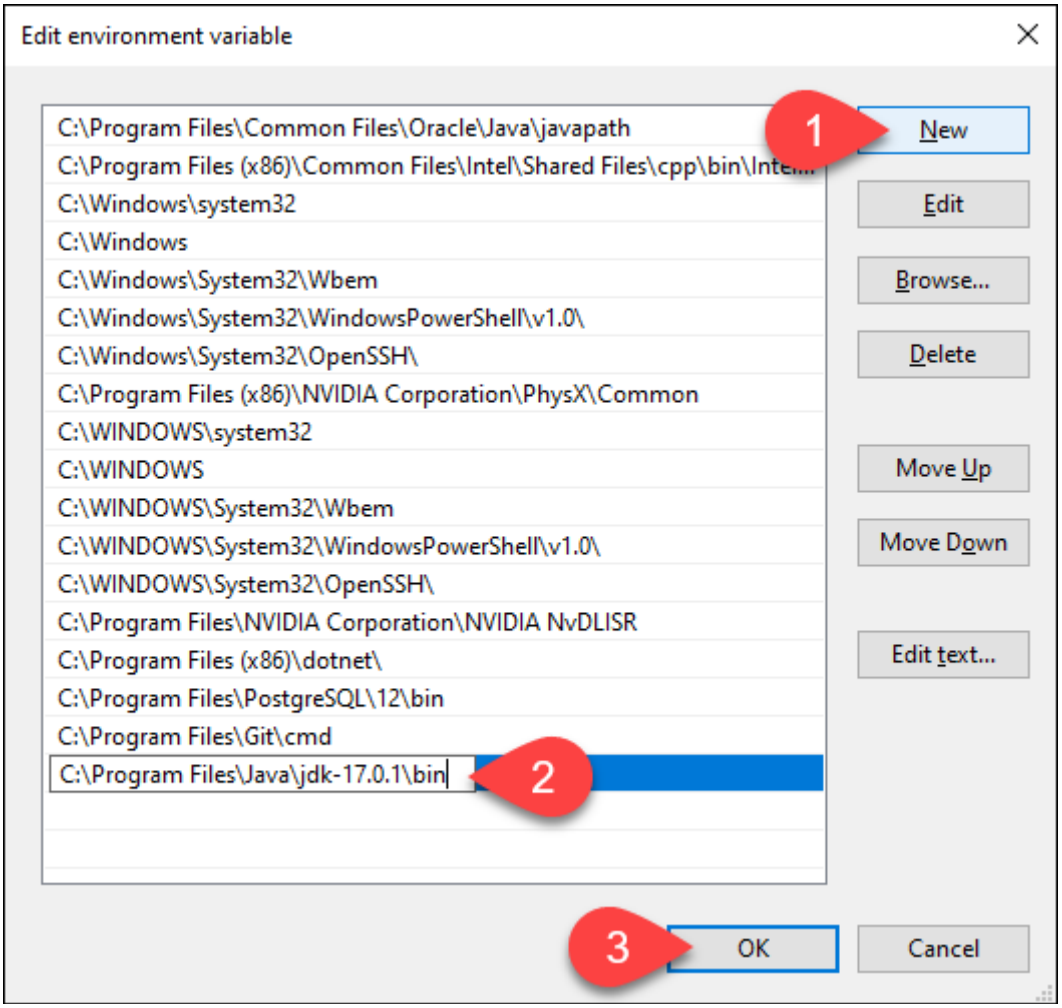

3. In the *System Properties* window, under the *Advanced* tab, click Environment Variables…




4. Under the *System variables* category, select the Path variable and click Edit:




5. Click the New button and enter the path to the Java bin directory:



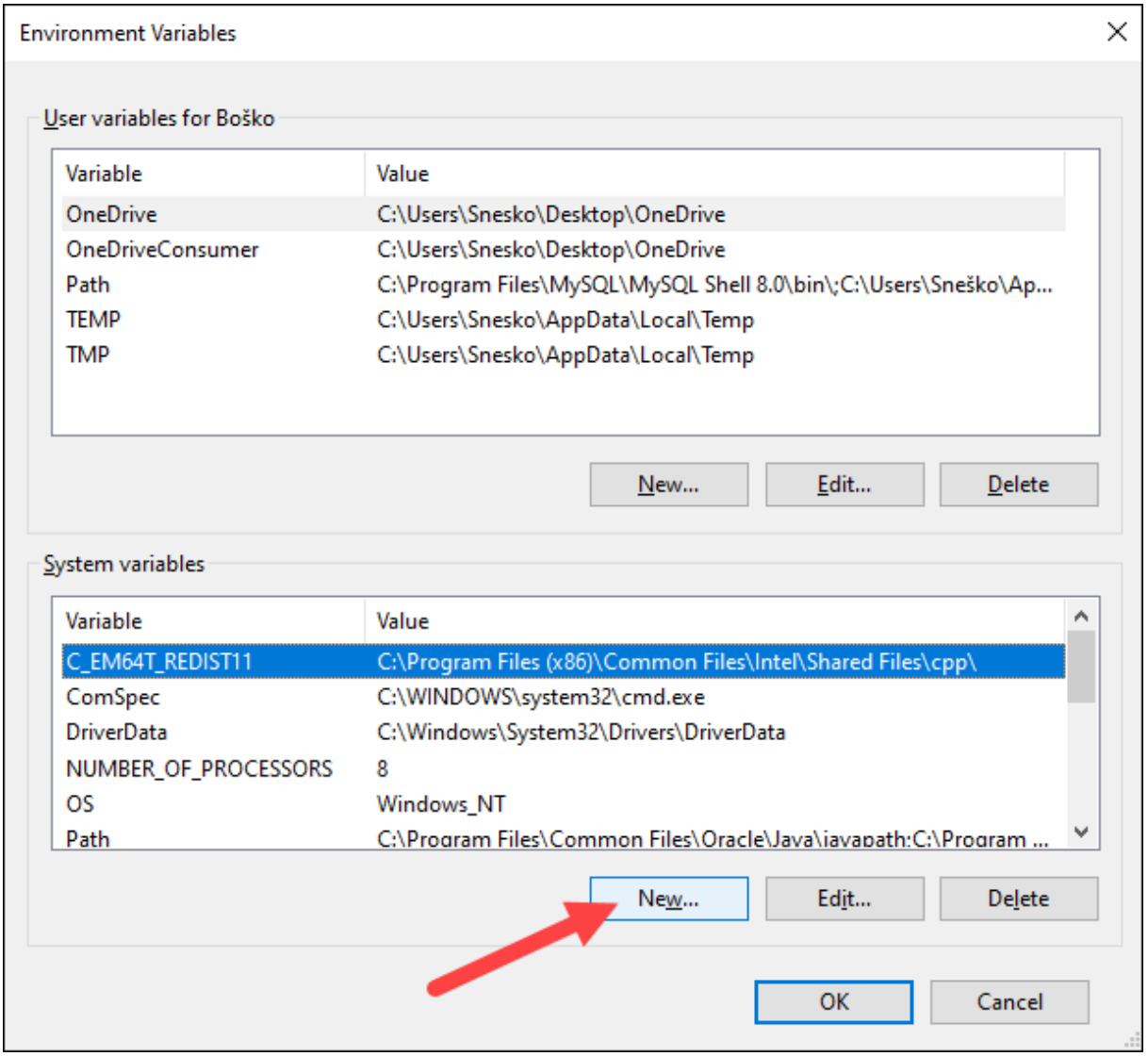

Note: The default path is usually *C:\Program Files\Java\jdk-17.0.1\bin*.

6. Click OK to save the changes and exit the variable editing window.

### Step 2: Add JAVA_HOME Variable

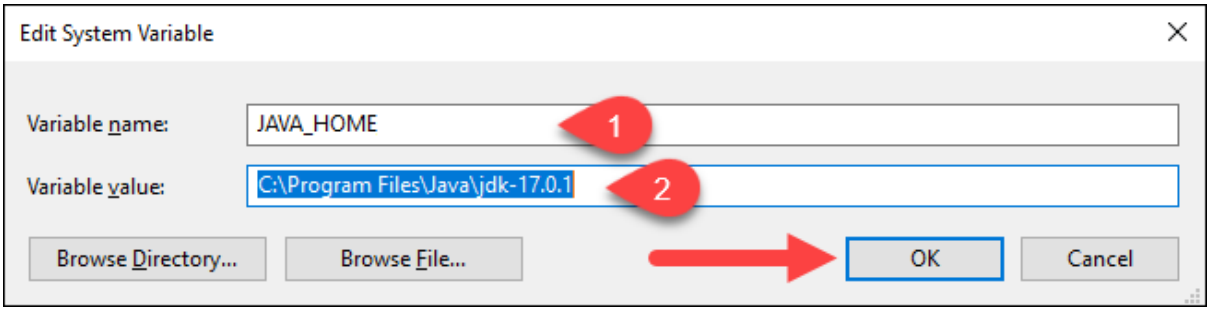
Some applications require the JAVA_HOME variable. Follow the steps below to create the variable:

1. In the *Environment Variables* window, under the *System variables* category, click the New… button to create a new variable.




2. Name the variable as *JAVA_HOME*.

3. In the variable value field, paste the path to your Java jdk directory and click OK.




**4. Confirm the changes by clicking OK in the *Environment Variables* and *System properties* windows.**

**Test the Java Installation**

Run the java -version command in the command prompt to make sure Java installed correctly:

```
C:\Users\boskom>java -version
java version "17.0.1" 2021-10-19 LTS
Java(TM) SE Runtime Environment (build 17.0.1+12-LTS-39)
Java HotSpot(TM) 64-Bit Server VM (build 17.0.1+12-LTS-39, mixed mode, sharing)
```

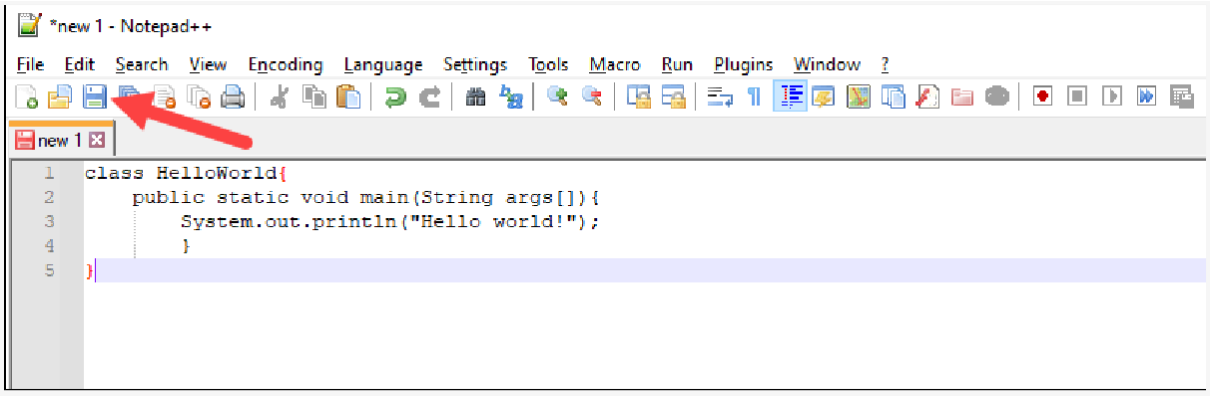
If installed correctly, the command outputs the Java version. Make sure everything works by writing a simple program and compiling it. Follow the steps below:

## Step 1: Write a Test Java Script

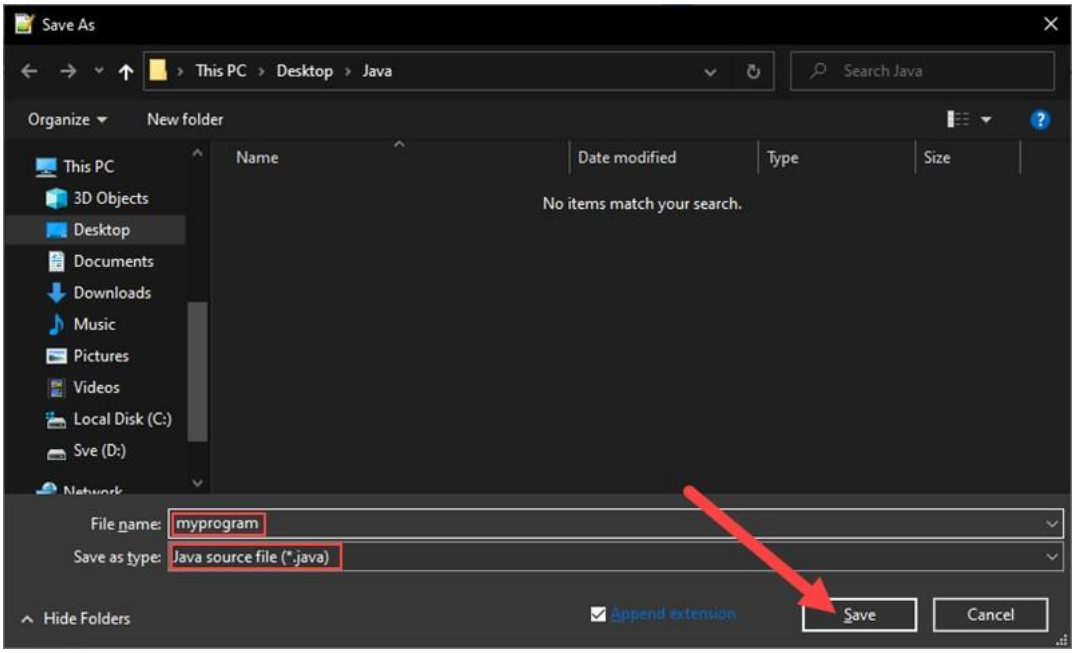
1. Open a text editor such as Notepad++ and create a new file.

2. Enter the following lines of code and click Save:

```
class HelloWorld{
        public static void main(String args[]){
                System.out.println("Hello world!");
                }
}
```




**3. Name the file and save it as a Java source file (*.java).**




Note: When using Notepad, select All files for the Save as type option and add the *.java* extension to the file name.

## Step 2: Compile the Test Java Script

1. In the command prompt, change the directory to the file's location and use the following syntax to compile the program:

**javac [filename]**
**For example:**

```
C:\Users\boskom\Desktop\Java>javac myprogram.java

C:\Users\boskom\Desktop\Java>
```

After a successful compilation, the program generates a *.class* file in the file directory.

2. Run the program with the following syntax:

**java [filename]**

```
C:\Users\boskom\Desktop\Java>java HelloWorld
Hello world!
```

The output shows that the program runs correctly, displaying the *Hello world!* Message.

# Installing the JDK on macOS

## System Requirements

The following are the system requirements for installing the JDK on macOS:

- Any Intel-based computer running macOS.
- Administrator privileges are required to install the JDK on macOS. We cannot install Java for a single user. Installing the JDK on macOS is performed on a systemwide basis for all users.

## Determining the Default JDK Version

- When starting a Java application through the command line, the system uses the default JDK.
- There can be multiple JDKs installed on the macOS system.
- You can determine which version of the JDK is the default by entering java -version in a Terminal window.
- If the installed version is 15 Interim 0, Update 0, and Patch 0, then you see a string that includes the text 15. For example:

```
$ java -version
java version "15"
Java(TM) SE Runtime Environment (build 15)
Java HotSpot(TM) 64-Bit Server VM (build 23.2-b04, mixed mode)
```

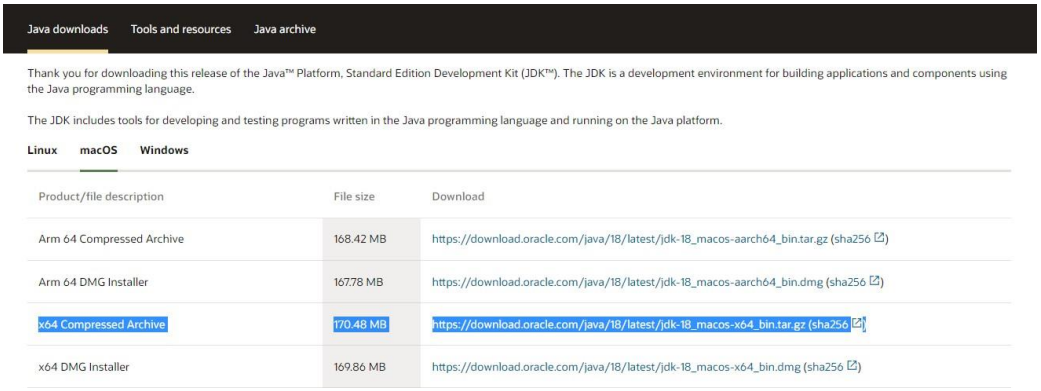
To run a different version of Java, either specify the full path, or use the java_home tool. For example:

```
$ /usr/libexec/java_home -v 15 --exec javac -version
```

## Installing the JDK

To install the JDK on macOS:

1. Download the JDK .dmg file, jdk-15.*interim.update.patch*_osx-x64_bin.dmg from Java SE Downloads page.
   Click **Accept License Agreement**.




2. From either the browser **Downloads** window or from the file browser, double-click the .dmg file to start it.
   A **Finder** window appears that contains an icon of an open box and the name of the .pkg file.
3. Double-click the JDK 15.pkg icon to start the installation application.
   The installation application displays the **Introduction** window.
4. Click **Continue**.
   The **Installation Type** window appears.
5. Click **Install**.
   A window appears that displays the message: Installer is trying to install new software. Enter your password to allow this.
6. Enter the Administrator user name and password and click **Install Software**.
   The software is installed and a confirmation window is displayed.

**Note:** After the software is installed, you can delete the .dmg file if you want to save disk space.

## Uninstalling the JDK on macOS

To uninstall the JDK on macOS:

- You must have Administrator privileges.
- **Note**: Do not attempt to uninstall Java by removing the Java tools from /usr/bin. This directory is part of the system software and any changes will be reset by Apple the next time you update the OS.

1. Go to /Library/Java/JavaVirtualMachines.
2. Remove the directory whose name matches the following format by executing the rm command as a root user or by using the sudo tool:

   Library/Java/JavaVirtualMachines/jdk-15.*interim.update.patch*.jdk

For example, to uninstall 15 Interim .

```
$ rm -rf jdk-15.jdk
```

# Installing JDK on Linux Platform

This topic describes general information about installing JDK 18 on Linux platforms.

Installing the JDK automatically creates a directory called

```
jdk-18.interim.update.patch
```

**Note**:If you install JDK in a specific location such as `/usr/jdk`, then you must log in with root credentials to gain the necessary permissions. If you do not have the root access, then either install the JDK in your home directory or a subdirectory for which you have the write permissions.

### Overwriting Files

If you install the software in a directory that contains a subdirectory named ***jdk-18.interim.update.patch***, then the new software overwrites files of the same name in that directory. Ensure that you rename the old directory if it contains files that you would like to keep.

### System Preferences

By default, the installation script configures the system such that the backing store for system preferences is created inside the JDK installation directory as applicable. If the JDK is installed on a network-mounted drive, then the system preferences can be exported for sharing with Java runtime environments on other machines.

## System Requirements for Installing JDK on Linux Platforms

### JDK Installation Instruction Notation for Linux Platforms

Instructions for installing JDK contain a version notation that represents the Feature, Interim, and Update version information.
For example, if you are installing JDK 18 Interim 0, Update 0, and Patch 0, then the following string represents the name of the bundle:

**Linux x64 systems:** *jdk-18.interim.update.patch_linux-x64_bin.tar.gz*
This string becomes:

```
jdk-18_linux-x64_bin.tar.gz
```

**Linux aarch64 (64-bit ARM) systems:** jdk-18.interim.update.patch_linux-aarch64_bin.tar.gz
This string becomes:

```
jdk-18_linux-aarch64_bin.tar.gz
```

***Note**:If a component has 0 as the suffix, it will be excluded*

## Installing the JDK on Linux from Archive Files and RPM Packages

- You can install the Java Development Kit (JDK) on a Linux platform from archive files or from Red Hat Package Manager (RPM) packages.

- Installation can be performed by using one of the following processes:
  - From archive files (.tar.gz): This allows you to install a private version of the JDK for the current user into any location, without affecting other JDK installations. However, it may involve manual steps to get some of the features to work. See the following table for the options available for downloading and installing the JDK from archive files. The bundles are available for Linux x64 and Linux aarch64 (64-bit ARM) systems. Install the required bundle.

  - From RPM packages (.rpm): This allows you to perform a systemwide installation of the JDK for all users, and requires root access. See the following table for the options available for downloading and installing the JDK from Linux x64 and Linux aarch64 (64-bit ARM) RPM packages.

- Downloading the JDK Installer

- Access Java SE Downloads page and click Accept License Agreement. Under the Download menu, click the Download link corresponding to your requirement.

- The following table lists the options and instructions for downloading and installing the JDK 18 release on a Linux platform:

| ***Download File*** | ***Instructions*** | ***Architecture (CPU Type)*** | ***Who Can Install*** |
|---|---|---|---|

| | | | |
|---|---|---|---|
| For Linux x64 systems: jdk-18.interim.update.patch_linux-x64.tar.gz | Installing the 64-Bit JDK on Linux Platforms | Intel - 64-bit | Anyone |
| For Linux aarch64 (64-bit ARM) systems: jdk-18.interim.update.patch_linux-aarch64.tar.gz | Installing the 64-Bit JDK on Linux Platforms | ARM - 64-bit | Anyone |
| For Linux x64 systems: jdk-18.interim.update.patch_linux-x64.rpm | Installing the 64-Bit JDK on RPM-Based Linux Platforms | Intel - 64-bit RPM-based Linux | Root |
| For Linux aarch64 (64-bit ARM) systems: jdk-18.interim.update.patch_linux-aarch64.rpm | Installing the 64-Bit JDK on RPM-Based Linux Platforms | ARM - 64-bit RPM-based Linux | Root |

## Installing the 64-Bit JDK on Linux Platforms

You can install the JDK for 64-bit Linux from an archive file (.tar.gz).

The .tar.gz archive file (also called a tarball) is a file that can be uncompressed and extracted in a single step.

To install the 64-bit JDK on a Linux platform:

1. Download the required file:
    - For Linux x64 systems: jdk-18.interim.update.patch_linux-x64_bin.tar.gz

- For Linux aarch64 (64-bit ARM) systems:
  jdk-18.interim.update.patch_linux-aarch64_bin.tar.gz

2. Before you download a file, you must accept the license agreement. Anyone (not only root users) can install the archive file in any location having write access.
3. Change the directory to the location where you want to install the JDK, then move the .tar.gz archive file to the current directory.
4. Unpack the tarball and install the downloaded JDK:
   $ tar zxvf jdk-18.interim.update.patch_linux-x64_bin.tar.gz
   OR
   *$ tar zxvf jdk-18.interim.update.patch_linux-aarch64_bin.tar.gz*
   The Java Development Kit files are installed in a directory called *jdk-18.interim.update.patch.*
5. Delete the .tar.gz file if you want to save disk space.

## Installing the 64-Bit JDK on RPM-Based Linux Platforms

You can install the JDK on 64-bit RPM-based Linux platforms, such as Oracle, Red Hat, and SuSE by using an RPM binary file (.rpm) in the system location.

Ensure that you have the root user access. You can do this by running the command su and entering the superuser password.

To install the 64-bit JDK on an RPM-based Linux platform:

1. Download the required file:
   - For x64 systems: j*dk-18.interim.update.patch_linux-x64_bin.rpm*
   - For aarch64 (64-bit ARM) systems:
     *jdk-18.interim.update.patch_linux-aarch64_bin.rpm*
2. Before you download a file, you must accept the license agreement.
3. Install the required package using the following command:
   *$ rpm -ivh jdk-18.interim.update.patch_linux-x64_bin.rpm*
   OR
   *$ rpm -ivh jdk-18.interim.update.patch_linux-aarch64_bin.rpm*

Note:JDK 18 can coexist with earlier versions of JDK. For each version, a new directory is created, the default directory being /usr/java/jdk-18.interim.update.patch.

1. Upgrade the required package using the following command:
   *$ rpm -Uvh jdk-18.interim.update.patch_linux-x64_bin.rpm*
   OR
   *$ rpm -Uvh jdk-18.interim.update.patch_linux-aarch64_bin.rpm*
2. Delete the .rpm file if you want to save disk space.
3. Exit the root shell.
   It is not required to reboot.

The JDK installation is integrated with the alternatives framework. After installation, the alternatives framework is updated to reflect the binaries from the recently installed JDK. Java commands such as java, javac, javadoc, and javap can be called from the command line.
Using the *java -version* command, users can confirm the default (recently installed) JDK version.
In addition, users can check which specific RPM package provides the java files:

```
$ rpm -q --whatprovides java
```