Before Proceeding with java, first we need to know what is a programming language?

A Programming language is an artificial language designed communicate instructions to a machine, particularly a computer. Programming languages can be used to create programs that control the behavior of a machine. In history there were many programming languages, in 1970, the most popular programming language was C. Since C was the most powerful language many people were using this language. This language was being used in business applications, scientific applications, artificial applications etc. Since, C has the procedure oriented programming features, it was having many limitations

Limitations of C are:

- Difficult to implement today's client requirements.
- No code reusability.
- No Code Enhancement
- Platform Dependent
- As length of the application increases, causes slow in performance.
- Code maintenance and enhancements are difficult.
- No Exception Handling mechanisms

In order to solve those disadvantages and new programming paradigm was introduce, named Object Oriented Programming(OOPs).

OOPs is basically based on objects and classes. OOPs can be defined as grouping of data and methods. OOPs basically have 4 features.

1. Inheritance
2. Polymorphism
3. Abstraction
4. Encapsulation

During the late 1970s and early 1980s, C became the dominant computer programming language, but in C complexity was more. The increasing complexity of programs has driven the need for better ways to manage that complexity. C++ is a response to that need. C++ was developed from C syntax, all the syntaxes were similar in C++ with C the only change was C++ was having OOPs features. By the end of the 1980s and the early 1990s, object-oriented programming using C++ took hold. C++ has overcome all the disadvantages of C, but still C++ was having its own disadvantages.

- Pointers.
- Explicitly memory allocation and de allocation.
- Platform Dependent.

Two Paradigms of Programming:

As you know, all computer programs consist of two elements: code and data. Further more,a program can be conceptually organized around its code or around its data. That is, some programs are written around “what is happening” and others are written around “who is being affected.” These are the two paradigms that govern how a program is constructed.

Procedure oriented Programming:

In this approach, the problem is always considered as a sequence of tasks to be done. A number of functions are written to accomplish these tasks. Here primary focus on “Functions” and little attention on data.

There are many high level languages like COBOL, FORTRAN, PASCAL, C used for conventional programming commonly known as POP.

Drawback: It does not model real world problems very well, because functions are action oriented and do not really corresponding to the elements of the problem.

Chars of POP:

- Emphasis is on doing actions.
- Large programs are divided into smaller programs known as functions.
- Most of the functions shared global data.
- Data move openly around the program from function to function.
- Functions transform data from one form to another.
- Employs top-down approach in program design.

OOP:

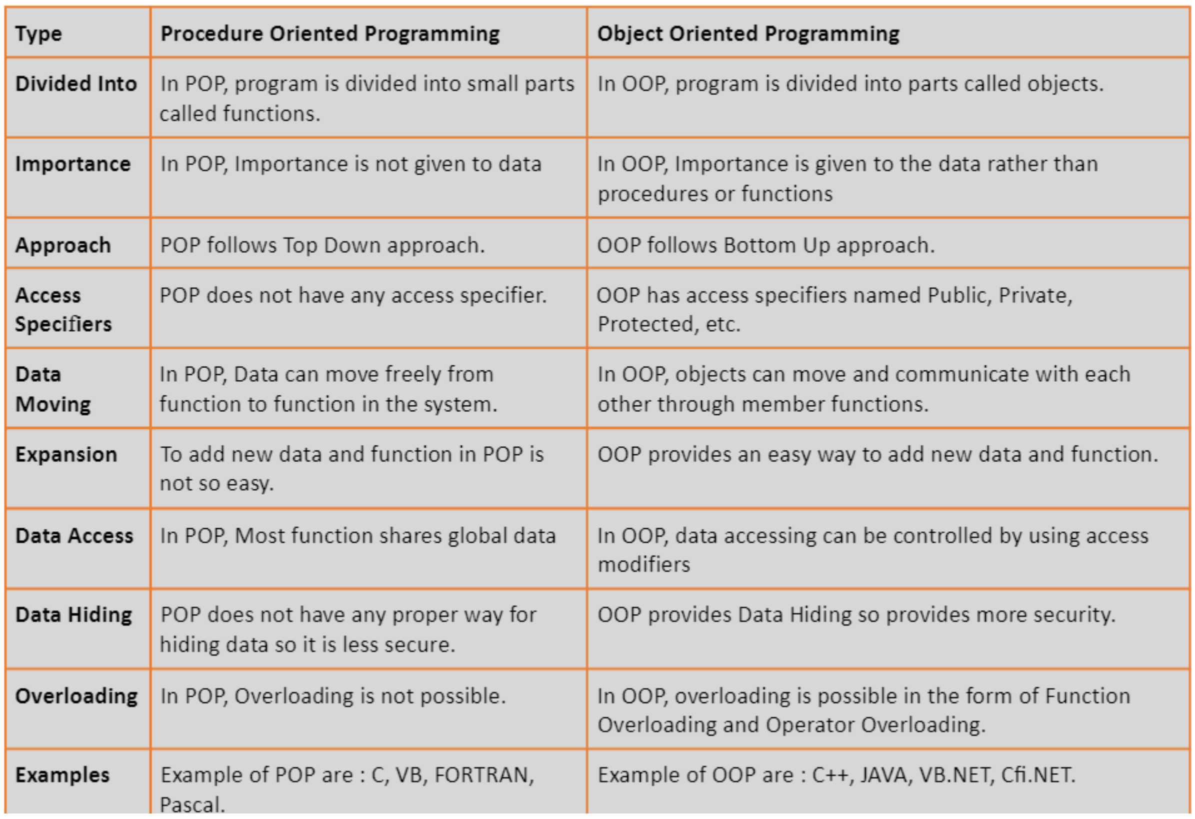
OOP allows us to decompose a problem into a number of entities called objects and then builds data and methods around these entities.

DEF: OOP is an approach that provides a way of modularizing programs by creating portioned memory area for both data and methods that can used as templates for creating copies of such modules on demand.

OOP Chars:

- Emphasis on data .
- Programs are divided into what are known as methods.
- Data structures are designed such that they characterize the objects.
- Methods that operate on the data of an object are tied together .
- Data is hidden.
- Objects can communicate with each other through methods.
- Reusability.

Follows bottom-up approach in program design.



Type	Procedure Oriented Programming	Object Oriented Programming
Divided Into	In POP, program is divided into small parts called functions.	In OOP, program is divided into parts called objects.
Importance	In POP, Importance is not given to data	In OOP, Importance is given to the data rather than procedures or functions
Approach	POP follows Top Down approach.	OOP follows Bottom Up approach.
Access Specifiers	POP does not have any access specifier.	OOP has access specifiers named Public, Private, Protected, etc.
Data Moving	In POP, Data can move freely from function to function in the system.	In OOP, objects can move and communicate with each other through member functions.
Expansion	To add new data and function in POP is not so easy.	OOP provides an easy way to add new data and function.
Data Access	In POP, Most function shares global data	In OOP, data accessing can be controlled by using access modifiers
Data Hiding	POP does not have any proper way for hiding data so it is less secure.	OOP provides Data Hiding so provides more security.
Overloading	In POP, Overloading is not possible.	In OOP, overloading is possible in the form of Function Overloading and Operator Overloading.
Examples	Example of POP are : C, VB, FORTRAN, Pascal.	Example of OOP are : C++, JAVA, VB.NET, Cfi.NET.

History Of java:

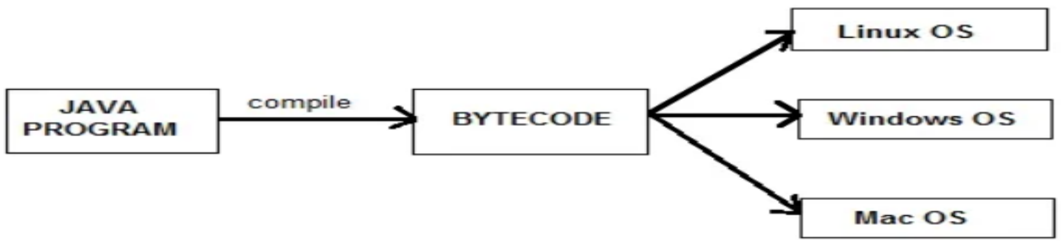
Java was created by James Gosling, Patrick Naughton, Chris Warth, Ed Frank, and Mike Sheridan at Sun Microsystems, Inc. in 1991. It took 18 months to develop the first working version. This language was initially called “Oak” but was renamed “Java” in 1995. Java’s primary motivation was the need for a platform-independent language that could be used to create software to be embedded in various consumer electronic devices, such as microwave ovens and remote controls. The trouble with C and C++ (and most other languages) is that they are designed to be compiled for a specific target. Although it is possible to compile a C++ program for just about any type of CPU, to do so requires a full C++ compiler targeted for that CPU. The problem is that compilers are expensive and time-consuming to create. An easier— and more cost-efficient—solution was needed. In an attempt to find such a solution, Gosling and others began work on a portable, platform-independent language that could be used to produce code that would run on a variety of CPUs under differing environments. This effort ultimately led to the creation of Java.

The Java Buzzwords(Features of Java)

- Simple
- Secure
- Portable
- Object-oriented
- Robust
- Multithreaded
- Architecture-neutral
- Interpreted
- High performance
- Distributed
- Dynamic

Platform independent:

Java is platform independent because java programs are executed by the any operating system irrespective of its development and compilation. Java program is compiled into bytecode. This bytecode is platform independent and can be run on any machine, plus this bytecode format also provide security. Any machine with Java Runtime Environment can run Java Programs.

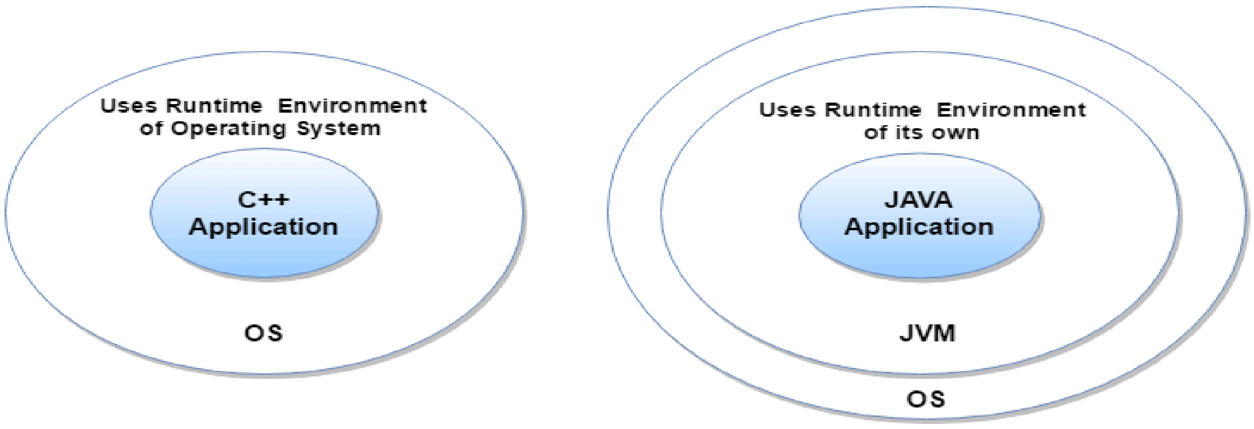


Simple:

There are various features that makes the java as a simple language. Programs are easy to write and debug because java does not use the pointers explicitly. The other reason is, if we know C and C++ then it is very easy to learn java, as we know all the syntaxes and rules of java are taken from C and C++. It also has the automatic memory allocation and deallocation system.

Secure:

Java provides internal and external security. Internal security is provided by the JVM(Java Virtual Machine). External security is provided by the developer using cryptographic techniques(Encryption, Decryption).


Uses Runtime Environment of Operating System
C++ Application
OS
Uses Runtime Environment of its own
JAVA Application
JVM
OS

Portable:

The feature **Write-once-run-anywhere** makes the java language portable. Byte code and JVM in java makes java as a platform independent. Java Byte code can be carried to any platform. No implementation dependent features. We have different versions of JVM for different platforms. Windows machines have their own version of JVM, Linux has its own and macOS has its own version of JVM. So if you distribute your bytecode to any machine, the JVM of that machine would translate the bytecode into the respective machine code.

Object-oriented:

Java is pure Object oriented programming language because everything we write in java comes under classes and objects. Java is pure object oriented programming language but it is not 100% object oriented programming language because java uses primitive data types which are defined using C and C++.

The main concepts of any Object Oriented Programming language are given below:

- ❖ Class and Object
- ❖ Encapsulation
- ❖ Abstraction

- Inheritance
- Polymorphism

Robust:

Java has the strong memory allocation and automatic garbage collection mechanism. It provides the powerful exception handling and type checking mechanism as compare to other programming languages. Compiler checks the program whether there any error and interpreter checks any run time error and makes the system secure from crash. All of the above features make the java language robust.

Multithreaded:

Java supports multithreaded programming, which allows you to write programs that do many things simultaneously. Thread is nothing but a piece of program and multithreading means executing more than one program simultaneously. For example consider an online examination application, in this application at one side one timer will be running and at the same time application will be displaying you question with options.

Architecture-neutral:

One of the main problems facing programmers is that no guarantee exists that if you write a program today, it will run tomorrow—even on the same machine. Operating system upgrades, processor upgrades, and changes in core system resources can all combine to make a program malfunction. But in java this problem was overcome. We can run java program anytime, anywhere, forever even though operating system, processor or other system resources upgrades.

Interpreted& High performance:

Java enables the creation of cross-platform programs by compiling into an intermediate representation called Java bytecode. This code can be executed on any system that implements the Java Virtual Machine. the Java bytecode was carefully designed so that it would be easy to translate directly into native machine code for very high performance by using a just-in-time compiler.

Distributed:

Java is designed for the distributed environment of the Internet, because it handles TCP/IP and HTTP protocols. Java achieves distributed features by RMI and by Socket Programming.

Dynamic:

Java is a dynamic language. It supports the dynamic loading of classes. It means classes are loaded on demand. It also supports functions from its native languages, i.e., C and C++.

Java Installation:

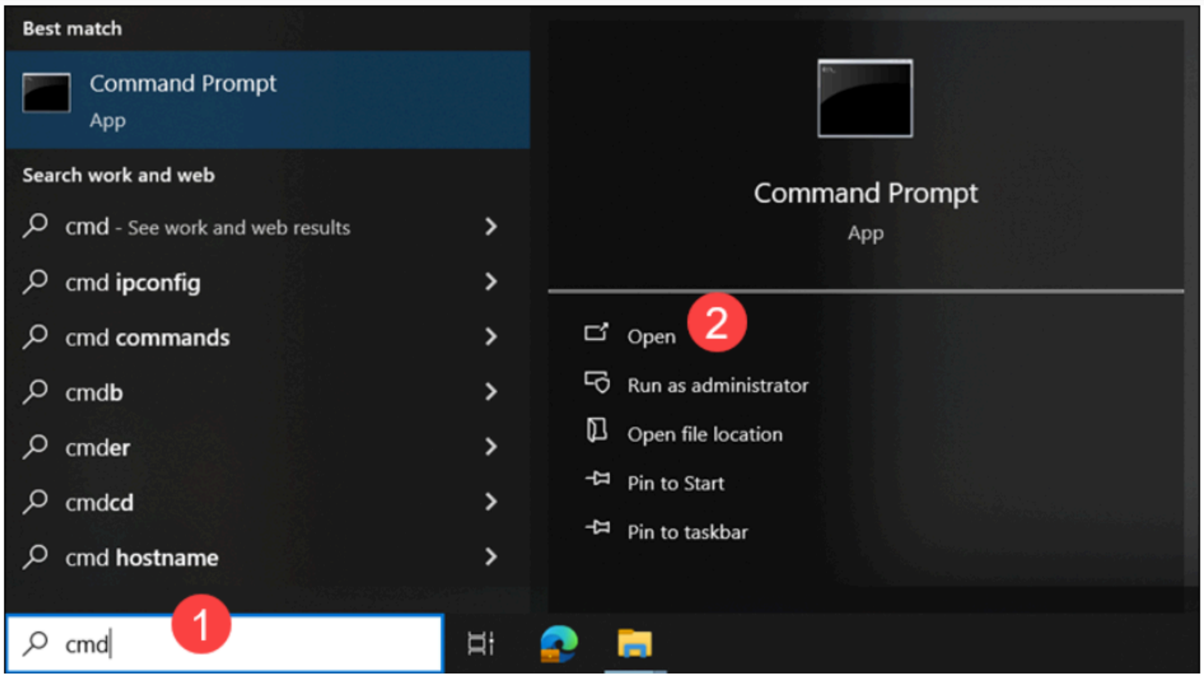
Check if Java Is Installed

Multiple Java versions on the same system can cause conflicts, as applications may attempt to use different versions. Additionally, outdated versions can pose significant security risks over time.

Before installing the latest Java Development Kit, check if a Java version is already installed on Windows:

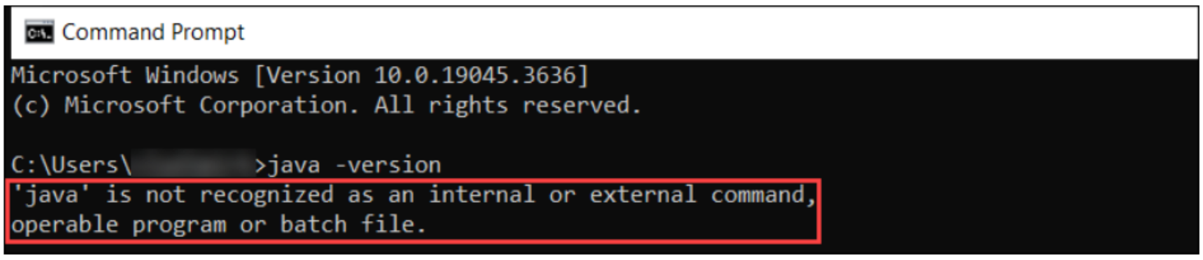
1. Type cmd in the Windows search bar.

2. Open the command prompt.



3. Enter the following command to check the Java version in Windows:

java -version



```
Command Prompt
Microsoft Windows [Version 10.0.19045.3636]
(c) Microsoft Corporation. All rights reserved.

C:\Users\[redacted]>java -version
'java' is not recognized as an internal or external command,
operable program or batch file.
```

In this example, the message states that *Java is not recognized as an internal or external command*, which indicates that Java is not installed. If the system displays a Java version number, remove the old Java installation before proceeding.

Download Java for Windows 10

OpenJDK is an open-source project that provides source code for implementing the Java platform. Many companies, including Oracle, offer distributions of OpenJDK in the form of installers or binaries.

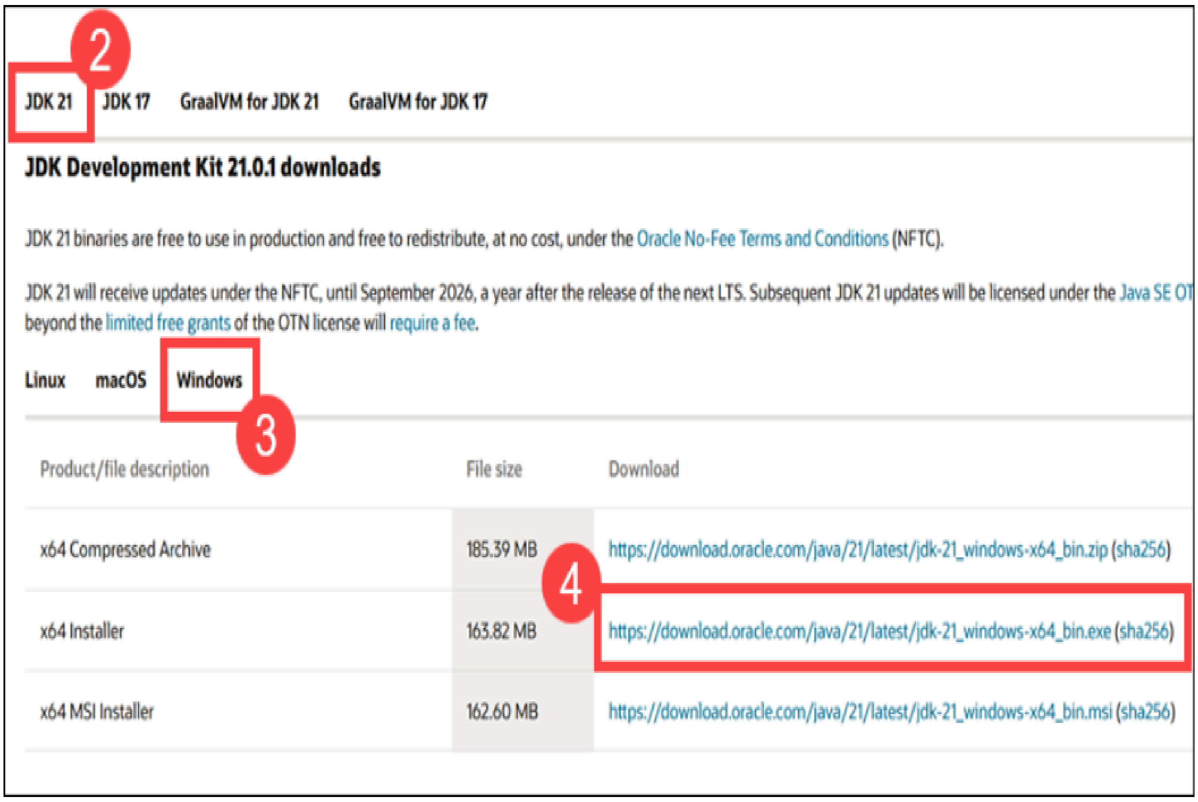
To download the latest Java Development Kit installation file for Windows 10:

1. Open a web browser and navigate to the Oracle Java Downloads page.

2. Select the latest JDK version. In this example, the latest available version is JDK 21.

3. Access the Windows tab.

4. Click the x64 Installer download link.



JDK Development Kit 21.0.1 downloads

JDK 21 binaries are free to use in production and free to redistribute, at no cost, under the Oracle No-Fee Terms and Conditions (NFTC).

JDK 21 will receive updates under the NFTC, until September 2026, a year after the release of the next LTS. Subsequent JDK 21 updates will be licensed under the Java SE OT
beyond the limited free grants of the OTN license will require a fee.

Product/file description	File size	Download
x64 Compressed Archive	185.39 MB	https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.zip (sha256)
x64 Installer	163.82 MB	https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.exe (sha256)
x64 MSI Installer	162.60 MB	https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.msi (sha256)

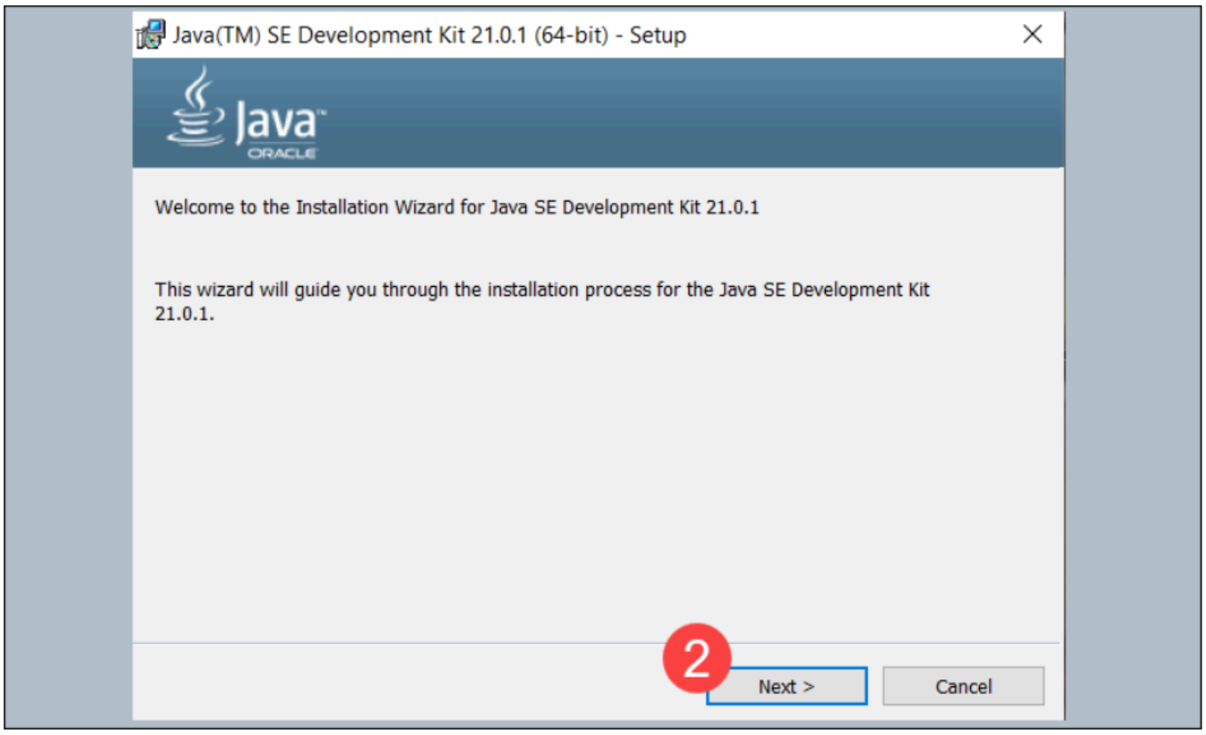
Wait for the download to complete.

Install Java on Windows 10

To install Java on your Windows system:

1. Double-click the **downloaded Java file** to start the installation.

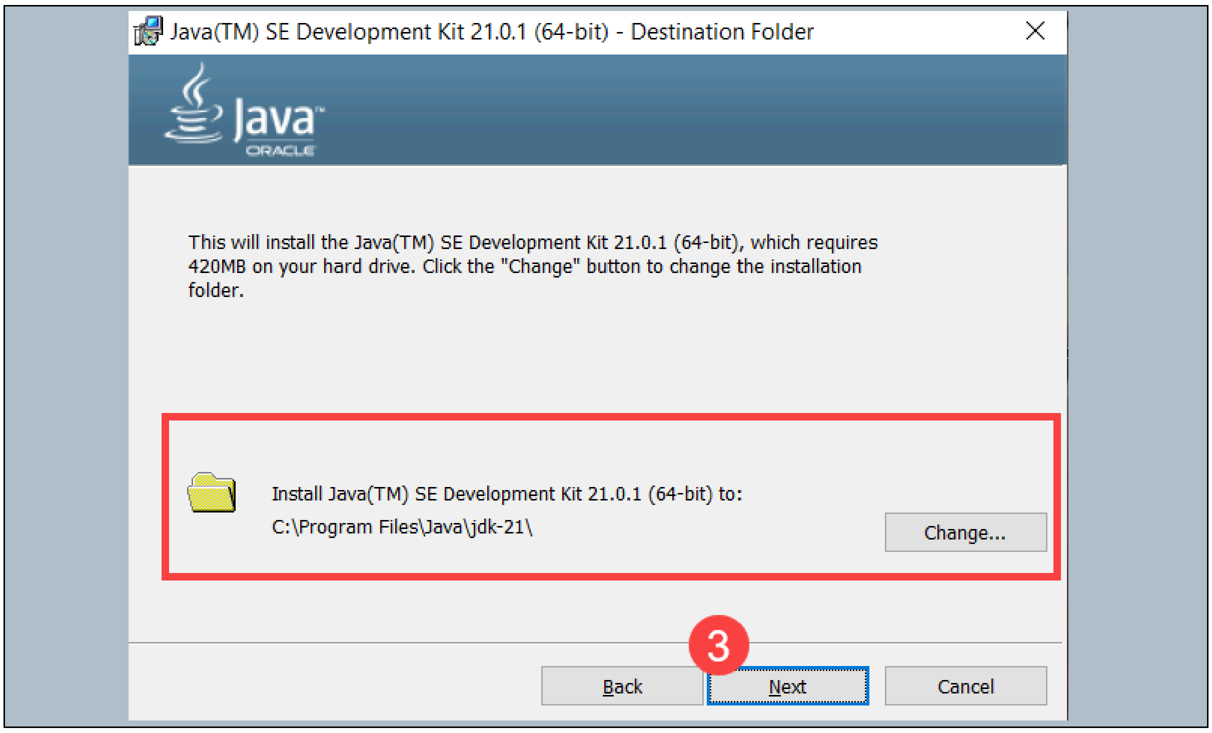
2. Once the installation wizard welcome screen appears, select **Next** to proceed.


Java(TM) SE Development Kit 21.0.1 (64-bit) - Setup

Welcome to the Installation Wizard for Java SE Development Kit 21.0.1

This wizard will guide you through the installation process for the Java SE Development Kit 21.0.1.

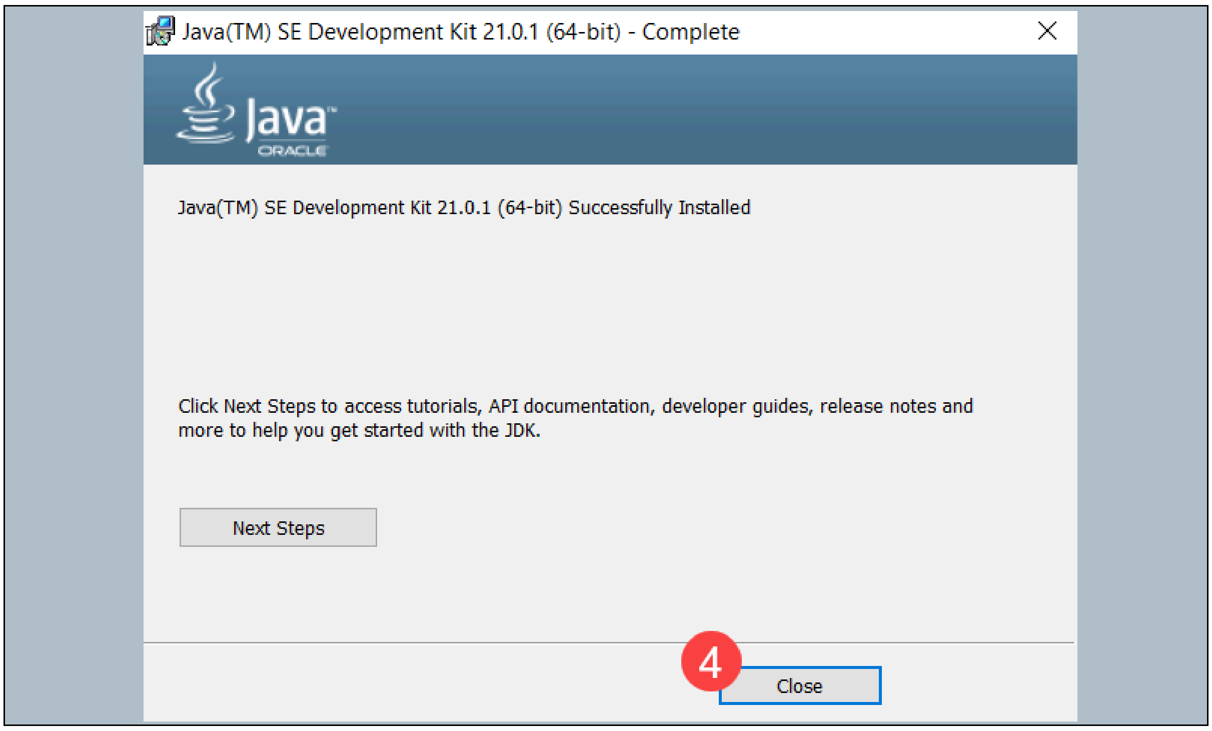
3. Choose the destination folder for the Java installation files, or stick to the default path and click **Next.**


Java(TM) SE Development Kit 21.0.1 (64-bit) - Destination Folder

This will install the Java(TM) SE Development Kit 21.0.1 (64-bit), which requires 420MB on your hard drive. Click the "Change" button to change the installation folder.

Install Java(TM) SE Development Kit 21.0.1 (64-bit) to:
C:\Program Files\Java\jdk-21\

4. The installation process is complete when the *Successfully Installed* message appears. Click **Close** to exit the wizard.


Java(TM) SE Development Kit 21.0.1 (64-bit) - Complete

Java(TM) SE Development Kit 21.0.1 (64-bit) Successfully Installed

Click Next Steps to access tutorials, API documentation, developer guides, release notes and more to help you get started with the JDK.

You have successfully installed JDK 21 on your Windows system. To enable program compiling from any directory, you must set up Java environment variables.

Set Environmental Variables in Java

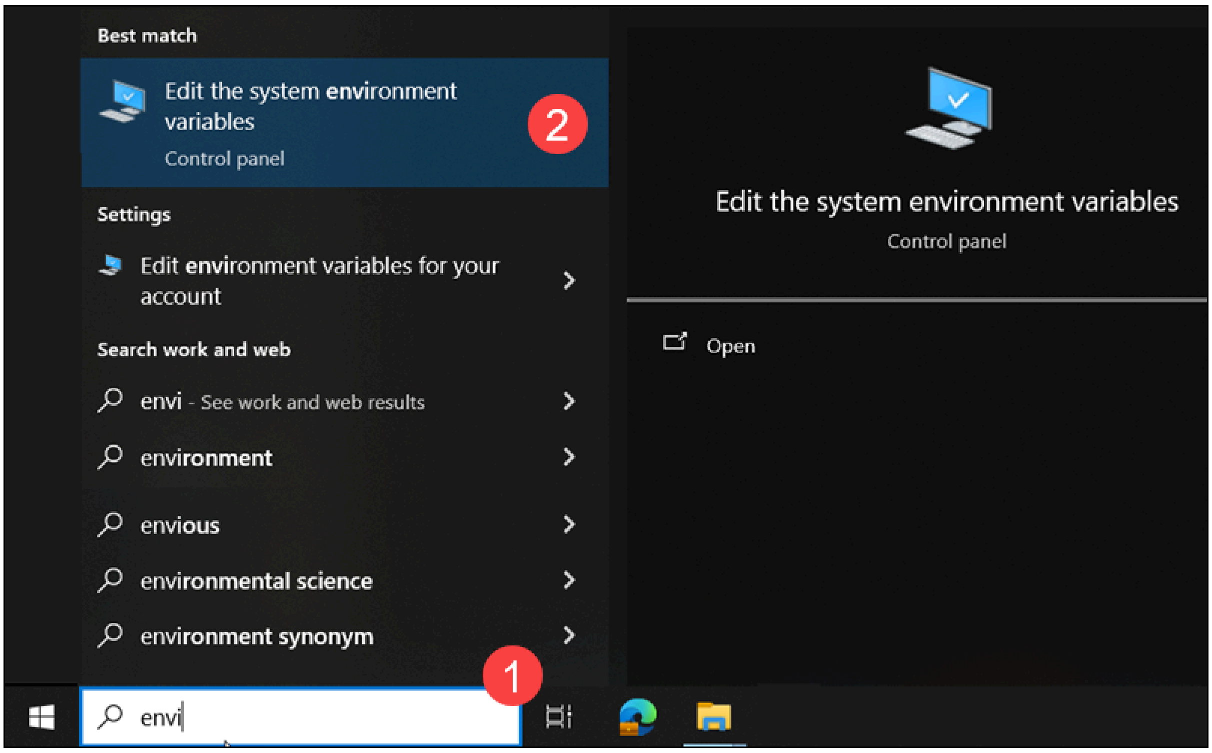
Follow the steps in the sections below to configure Java environment variables in Windows.

Step 1: Add Java to System Variables

This step ensures that Java is accessible from the command line in any directory.

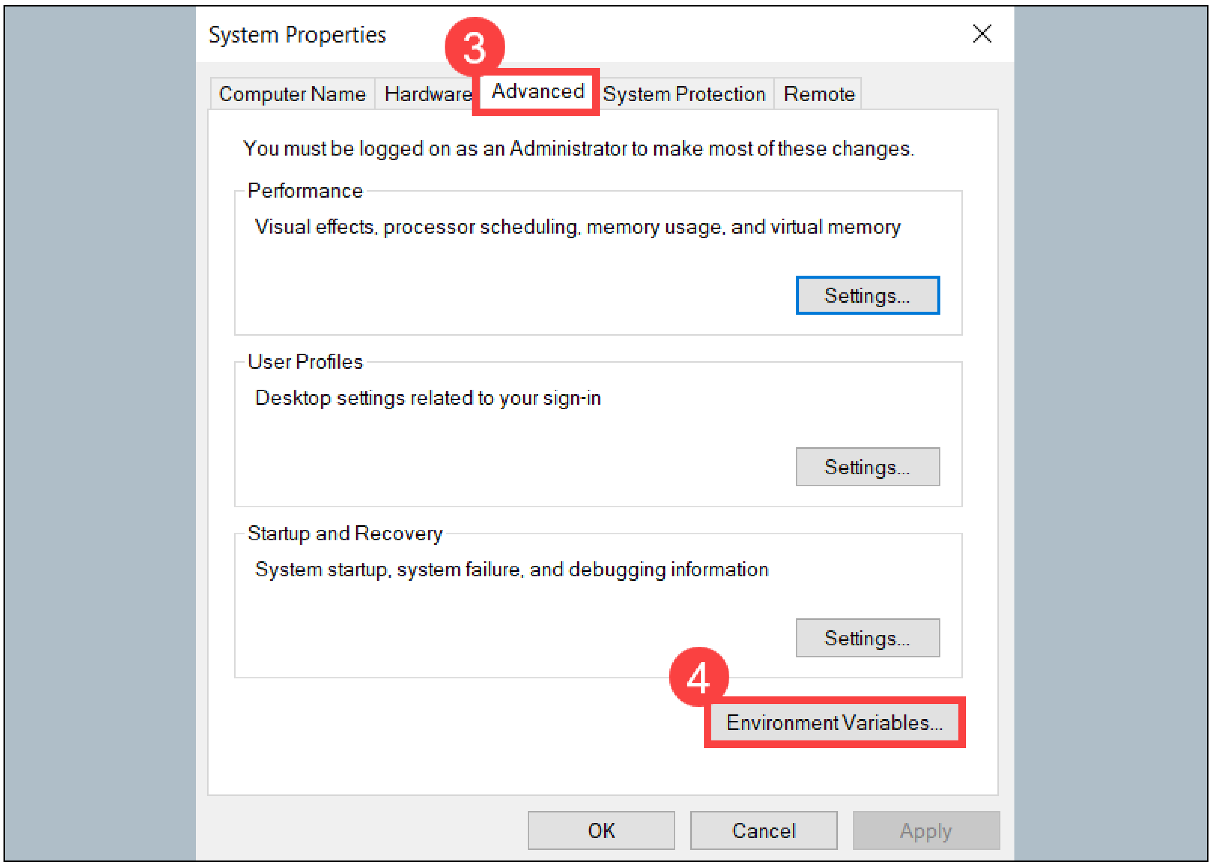
1. Open the **Start** menu and search for *environment variables*.

2. Select **Edit the system environment variables**.


Best match
Edit the system environment variables
Control panel
2
Settings
Edit environment variables for your account
Search work and web
envi - See work and web results
environment
envious
environmental science
environment synonym
1
envi
Edit the system environment variables
Control panel
Open

3. Select **Advanced** in the *System Properties* window.

4. Click **Environment Variables.**



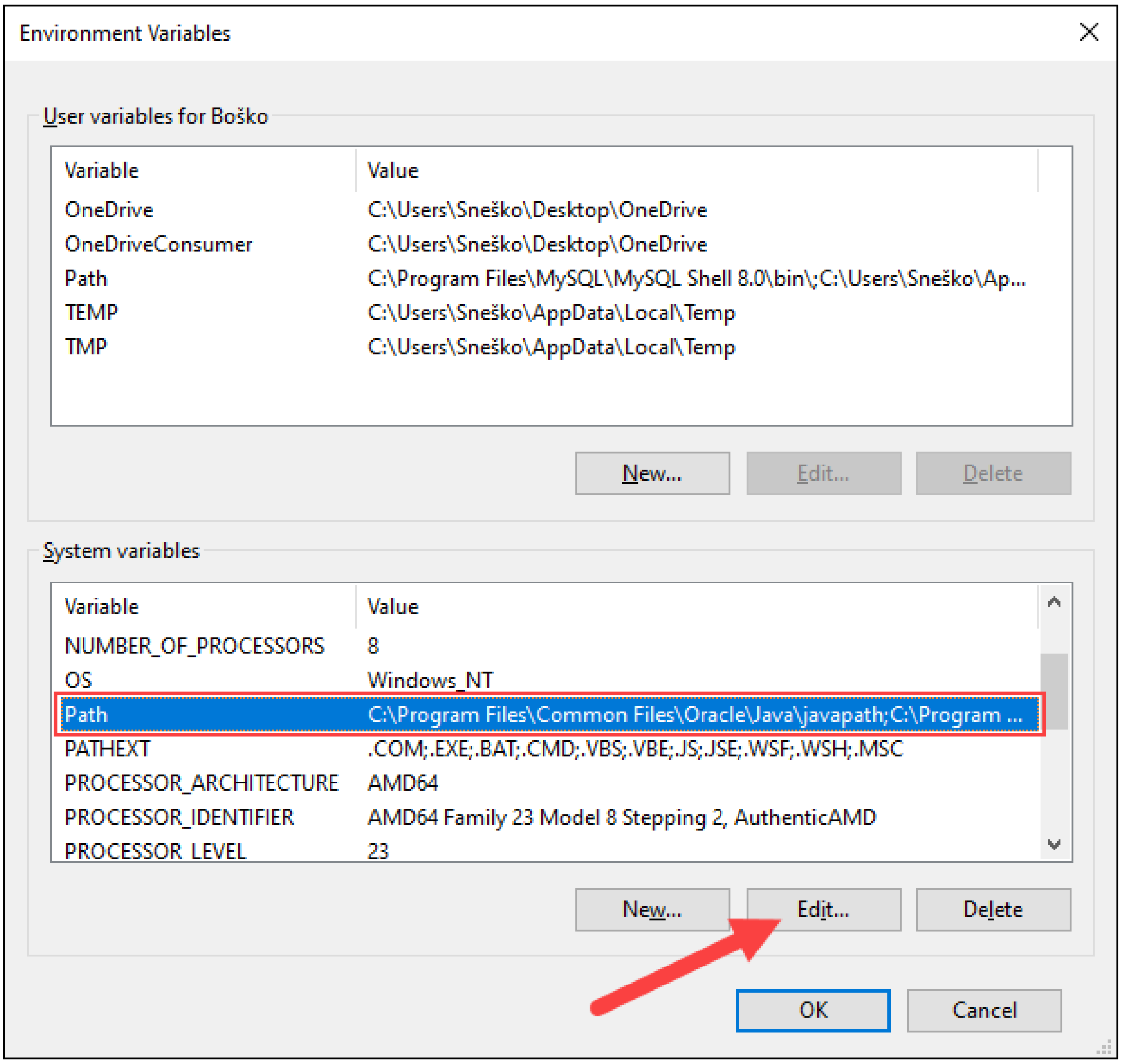
You must be logged on as an Administrator to make most of these changes.

Visual effects, processor scheduling, memory usage, and virtual memory

Desktop settings related to your sign-in

System startup, system failure, and debugging information

5. Select the **Path** variable in the *System variables* category and click **Edit**.


Environment Variables

User variables for Boško

Variable	Value
OneDrive	C:\Users\Sneško\Desktop\OneDrive
OneDriveConsumer	C:\Users\Sneško\Desktop\OneDrive
Path	C:\Program Files\MySQL\MySQL Shell 8.0\bin\;C:\Users\Sneško\Ap...
TEMP	C:\Users\Sneško\AppData\Local\Temp
TMP	C:\Users\Sneško\AppData\Local\Temp

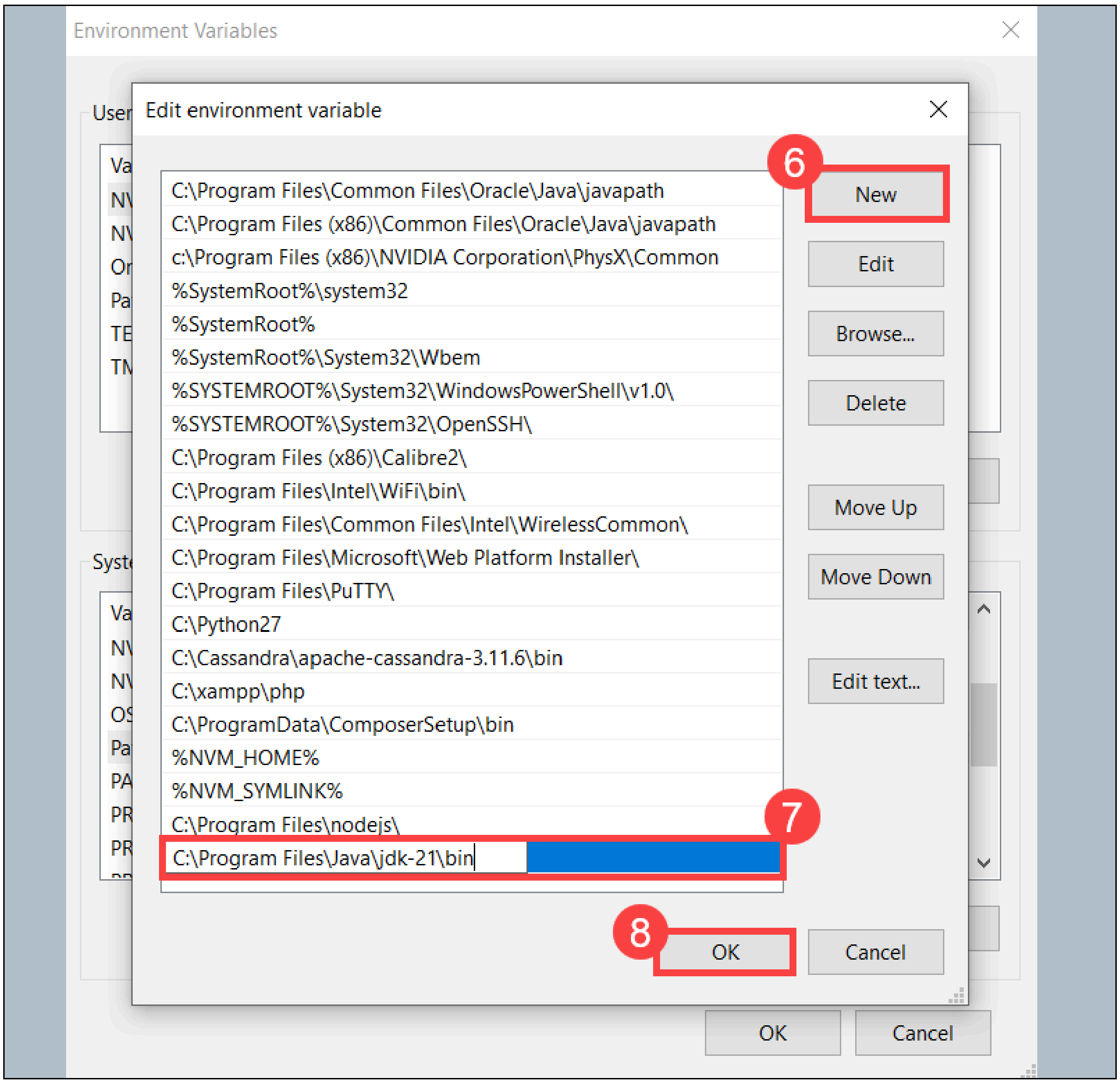
System variables

Variable	Value
NUMBER_OF_PROCESSORS	8
OS	Windows_NT
Path	C:\Program Files\Common Files\Oracle\Java\javapath;C:\Program ...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64
PROCESSOR_IDENTIFIER	AMD64 Family 23 Model 8 Stepping 2, AuthenticAMD
PROCESSOR_LEVEL	23

6. Click **New**.

7. Enter the path to the Java bin directory.

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


Environment Variables

Edit environment variable

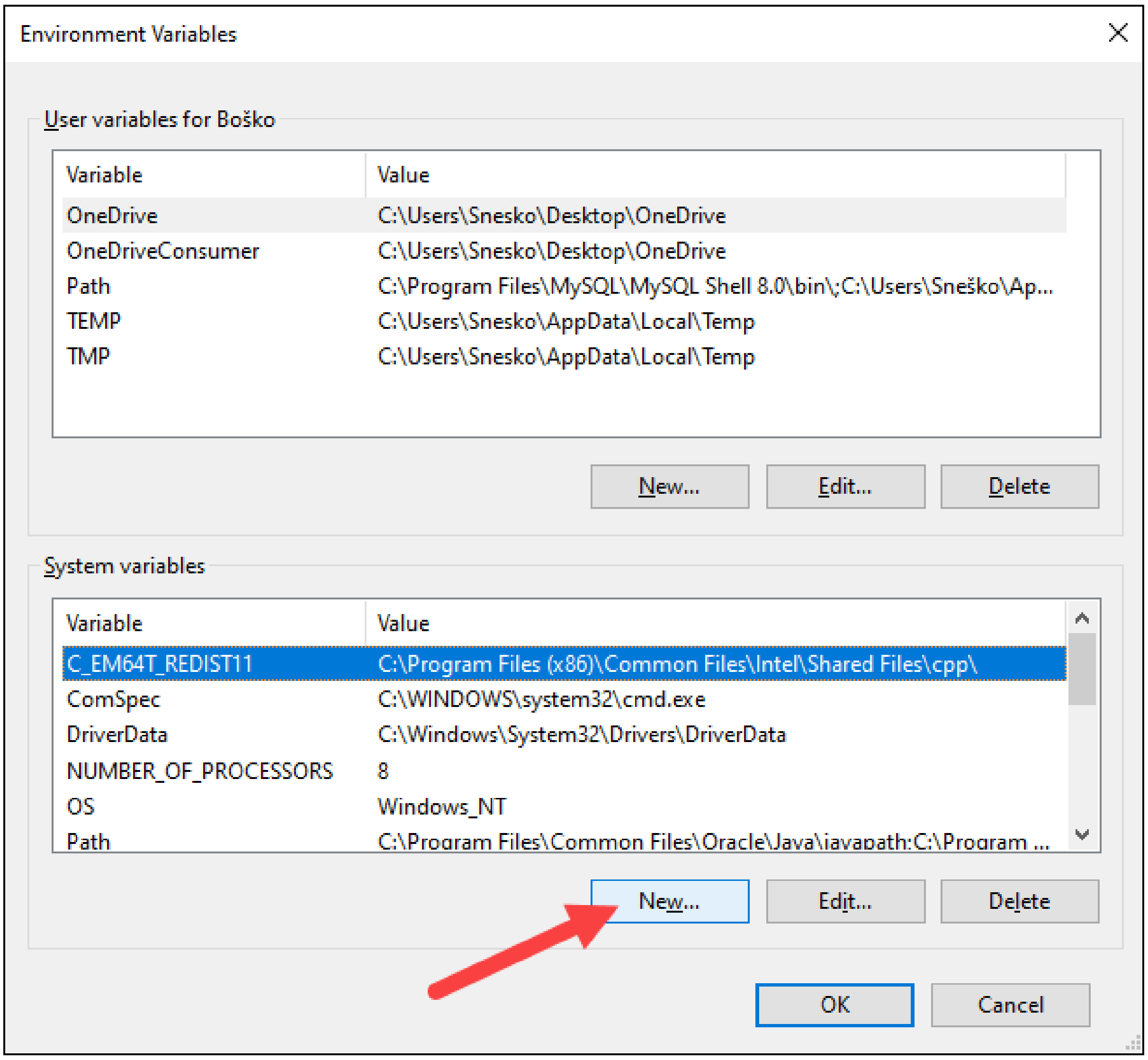
C:\Program Files\Common Files\Oracle\Java\javapath
C:\Program Files (x86)\Common Files\Oracle\Java\javapath
c:\Program Files (x86)\NVIDIA Corporation\PhysX\Common
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\Program Files (x86)\Calibre2\
C:\Program Files\Intel\WiFi\bin\
C:\Program Files\Common Files\Intel\WirelessCommon\
C:\Program Files\Microsoft\Web Platform Installer\
C:\Program Files\PuTTY\
C:\Python27
C:\Cassandra\apache-cassandra-3.11.6\bin
C:\xampp\php
C:\ProgramData\ComposerSetup\bin
%NVM_HOME%
%NVM_SYMLINK%
C:\Program Files\nodejs\
C:\Program Files\Java\jdk-21\bin

Note: The default path for **JDK 21** is *C:\Program Files\Java\jdk-21\bin*.

Step 2: Add JAVA_HOME Variable

Some applications require the **JAVA_HOME** variable to point to the JDK installation directory. Follow the steps below to create the variable:

1. Click **New** under the *System variables* category to create a new variable.


Environment Variables

User variables for Boško

Variable	Value
OneDrive	C:\Users\Snesko\Desktop\OneDrive
OneDriveConsumer	C:\Users\Snesko\Desktop\OneDrive
Path	C:\Program Files\MySQL\MySQL Shell 8.0\bin\;C:\Users\Sneško\Ap...
TEMP	C:\Users\Snesko\AppData\Local\Temp
TMP	C:\Users\Snesko\AppData\Local\Temp

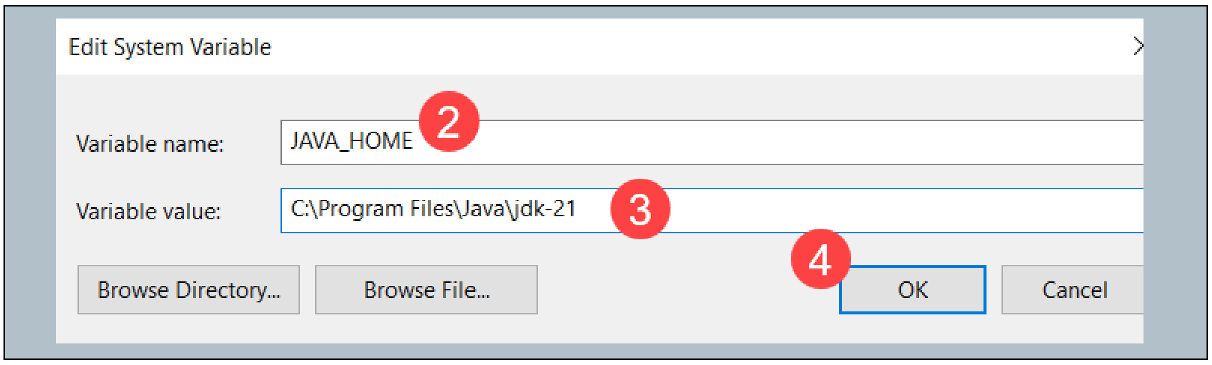
System variables

Variable	Value
C_EM64T_REDIST11	C:\Program Files (x86)\Common Files\Intel\Shared Files\cpp\
ComSpec	C:\WINDOWS\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	8
OS	Windows_NT
Path	C:\Program Files\Common Files\Oracle\Java\javapath;C:\Program ...

2. Name the variable **JAVA_HOME**.

3. Enter the path to your Java JDK directory in the variable value field.

4. Click **OK**.


Edit System Variable

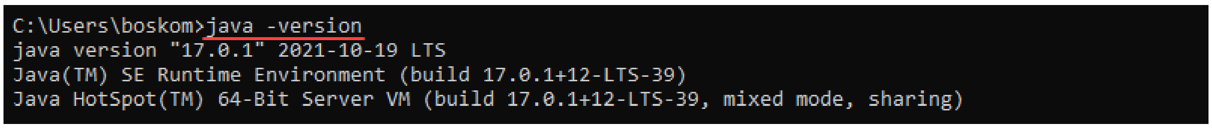
Variable name: JAVA_HOME

Variable value: C:\Program Files\Java\jdk-21

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

Test Java Installation

Verify that Java is installed by entering the **java -version** command in the command prompt:



```
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)
```

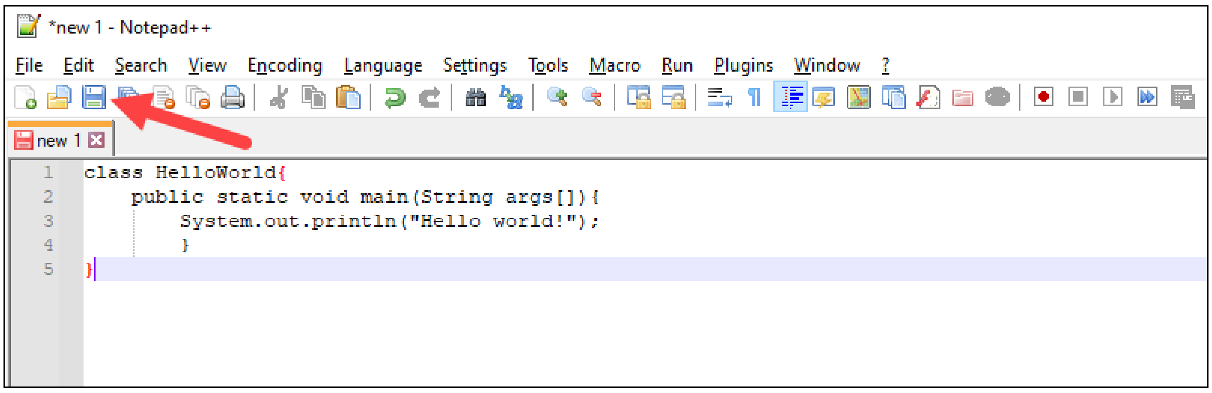
If installed correctly, the command outputs the Java version. To ensure everything works, write and compile a simple Java program by following the steps in the sections below.

Step 1: Write Test Java Script

1. Open a text editor like *Notepad* or *Notepad++* and create a new file.

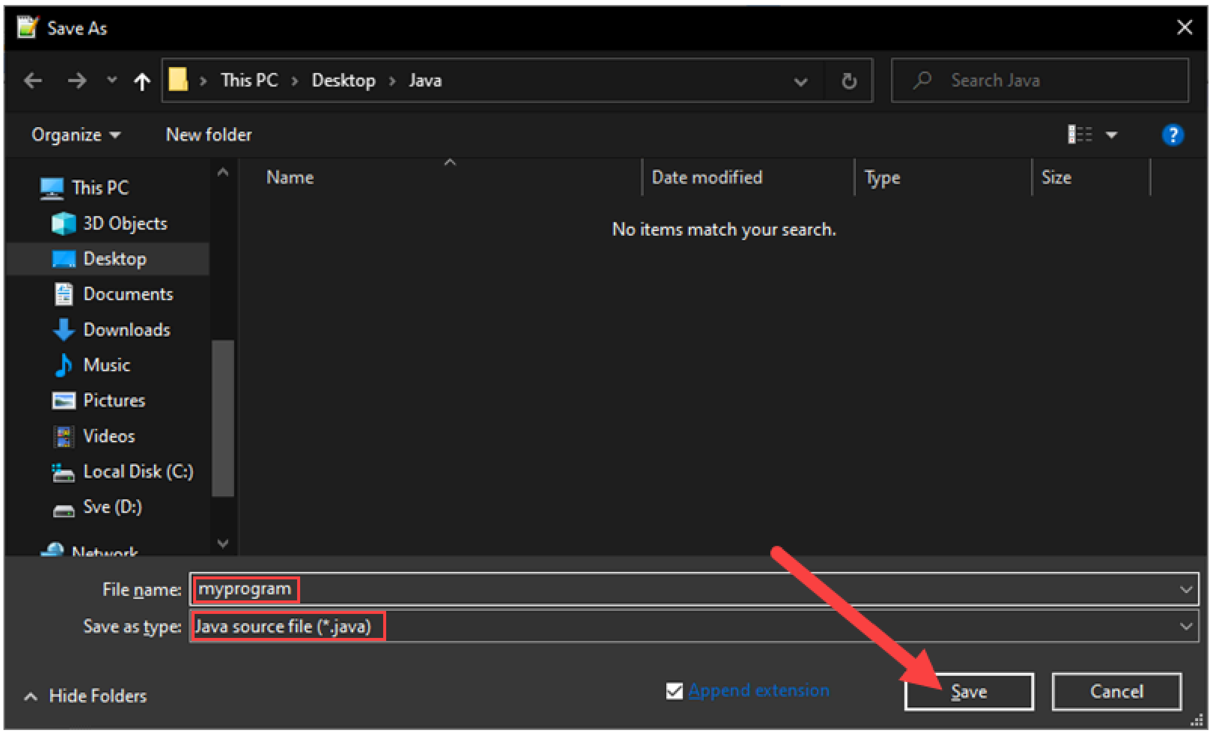
2. Enter the following code and click **Save**:

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



```
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)**.



No items match your search.

File name: myprogram

Save as type: 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 Test Java Script

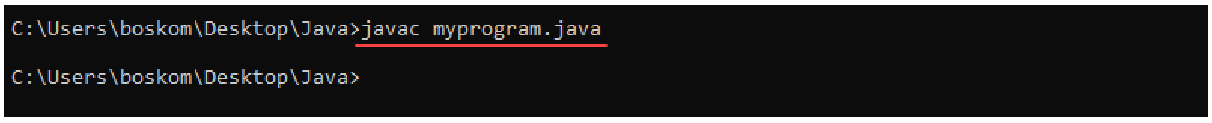
Access the Windows command prompt and complete the following steps:

1. Navigate to the directory where your Java file is saved.

2. Use the following command to compile the program:

javac myprogram.java

Replace **myprogram.java** with your file name.



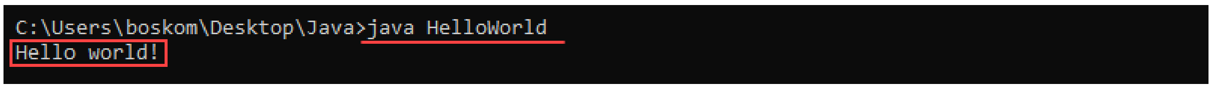
```
C:\Users\boskom\Desktop\Java>javac myprogram.java

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

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

2. Run the program using the following syntax:

java HelloWorld



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

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