Java Programs

Lecture (1)

By
Dr. Radwa & Dr. Ghada Fathy

Table of contents

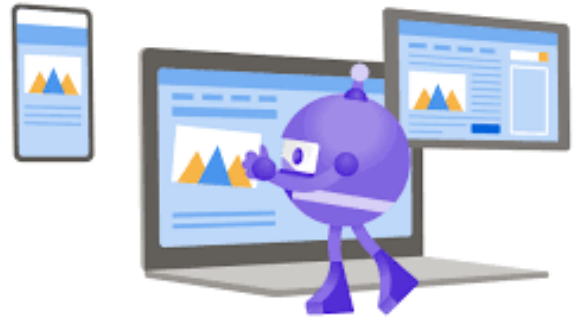
What is Java?

Java is a popular programming language, created in 1995.

It is owned by Oracle, and more than 3 billion devices run Java.

It is used for:

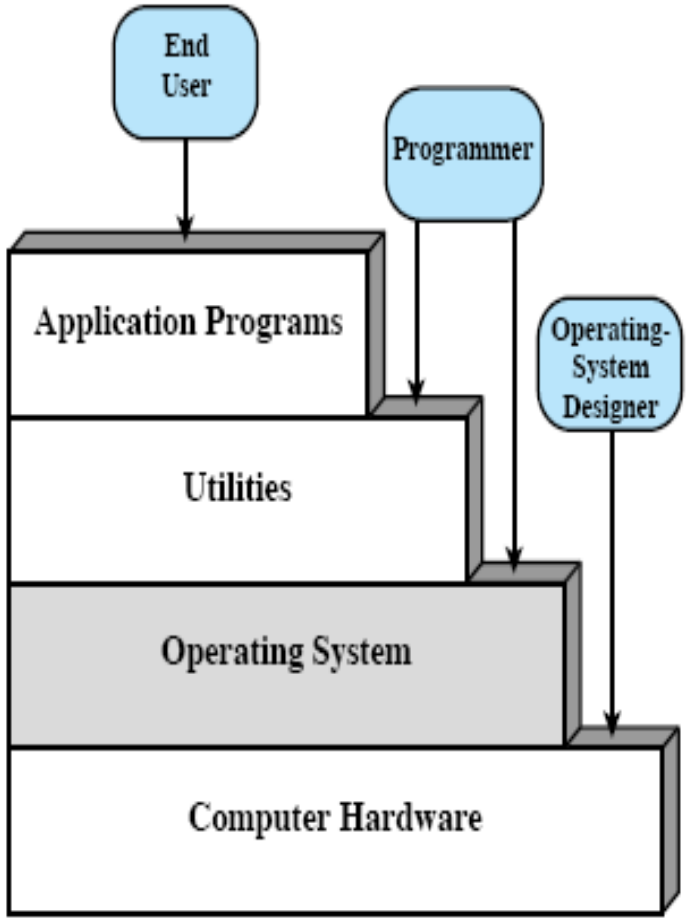
- Mobile applications (specially Android apps)
- Desktop applications
- Web applications
- Web servers and application servers
- Games
- Database connection
- And much, much more!

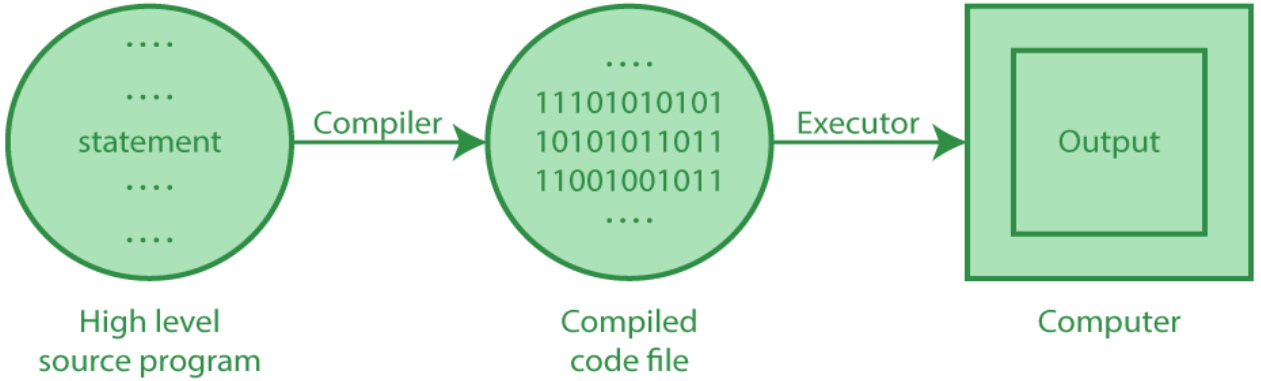


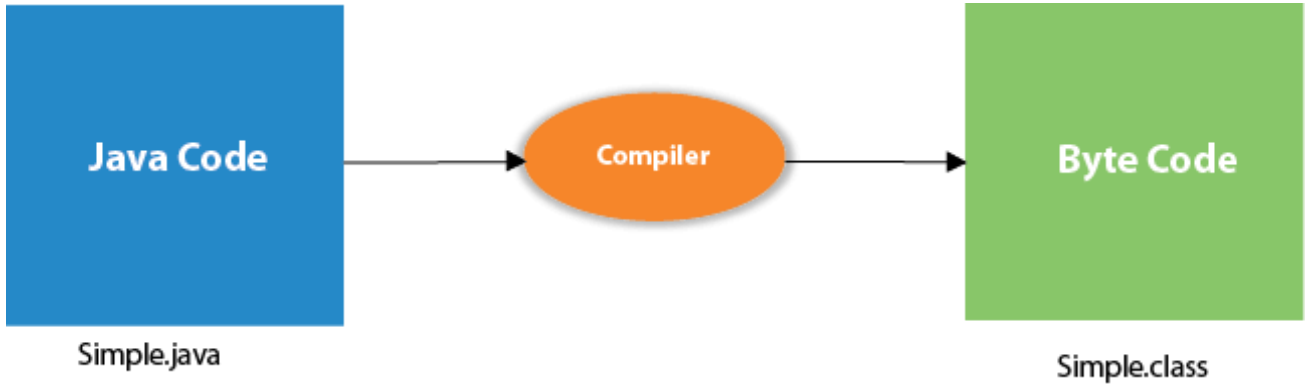
Why Use Java?

- Java works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.)
- It is one of the most popular programming language in the world
- It has a large demand in the current job market
- It is easy to learn and simple to use
- It is open-source and free
- It is secure, fast and powerful
- It has a huge community support (tens of millions of developers)
- Java is an **object oriented** language which gives a clear structure to programs and allows code to be reused, lowering development costs.
- As Java is close to C++ and C#, it makes it easy for programmers to switch to Java or vice versa

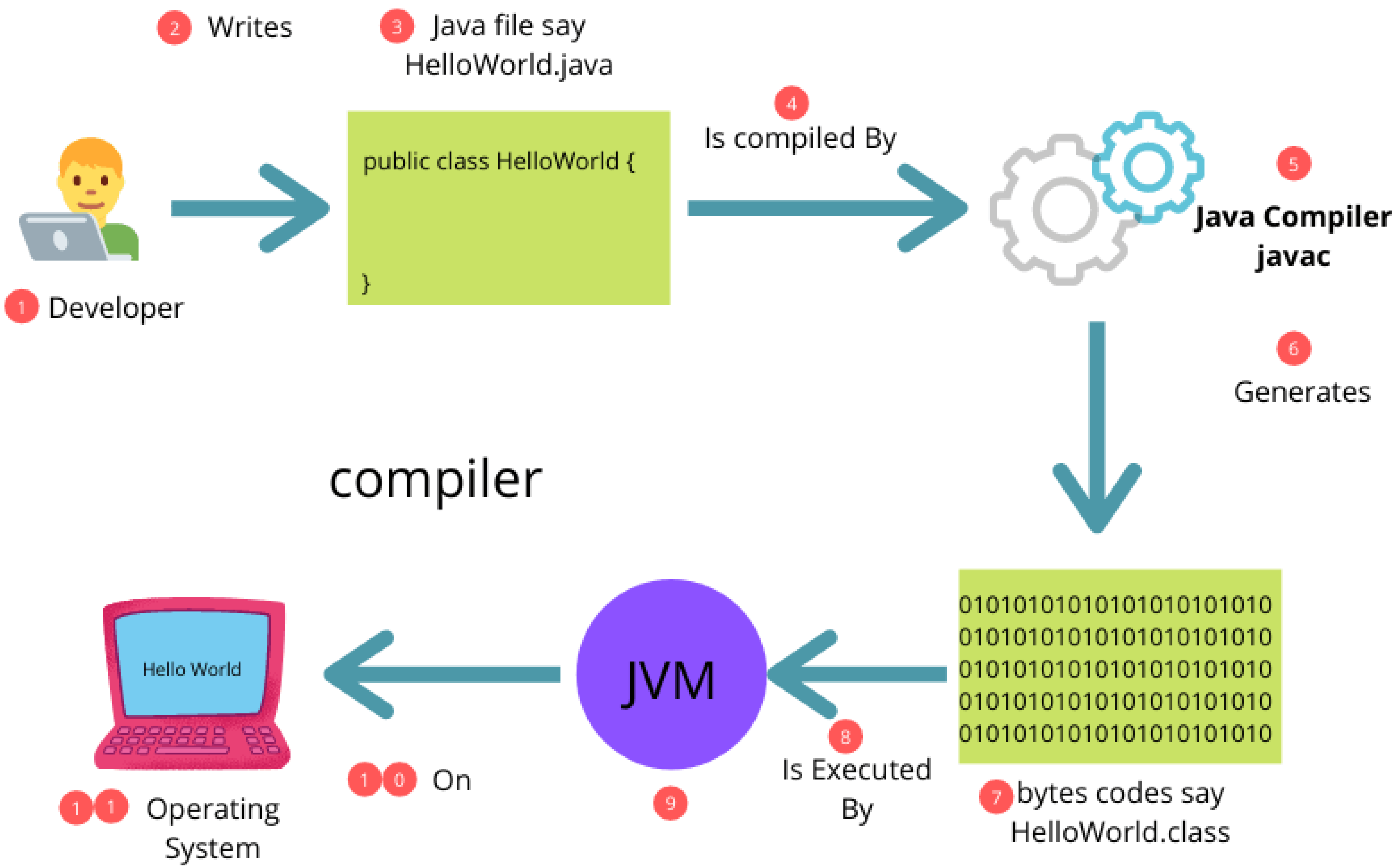
How Java Work?



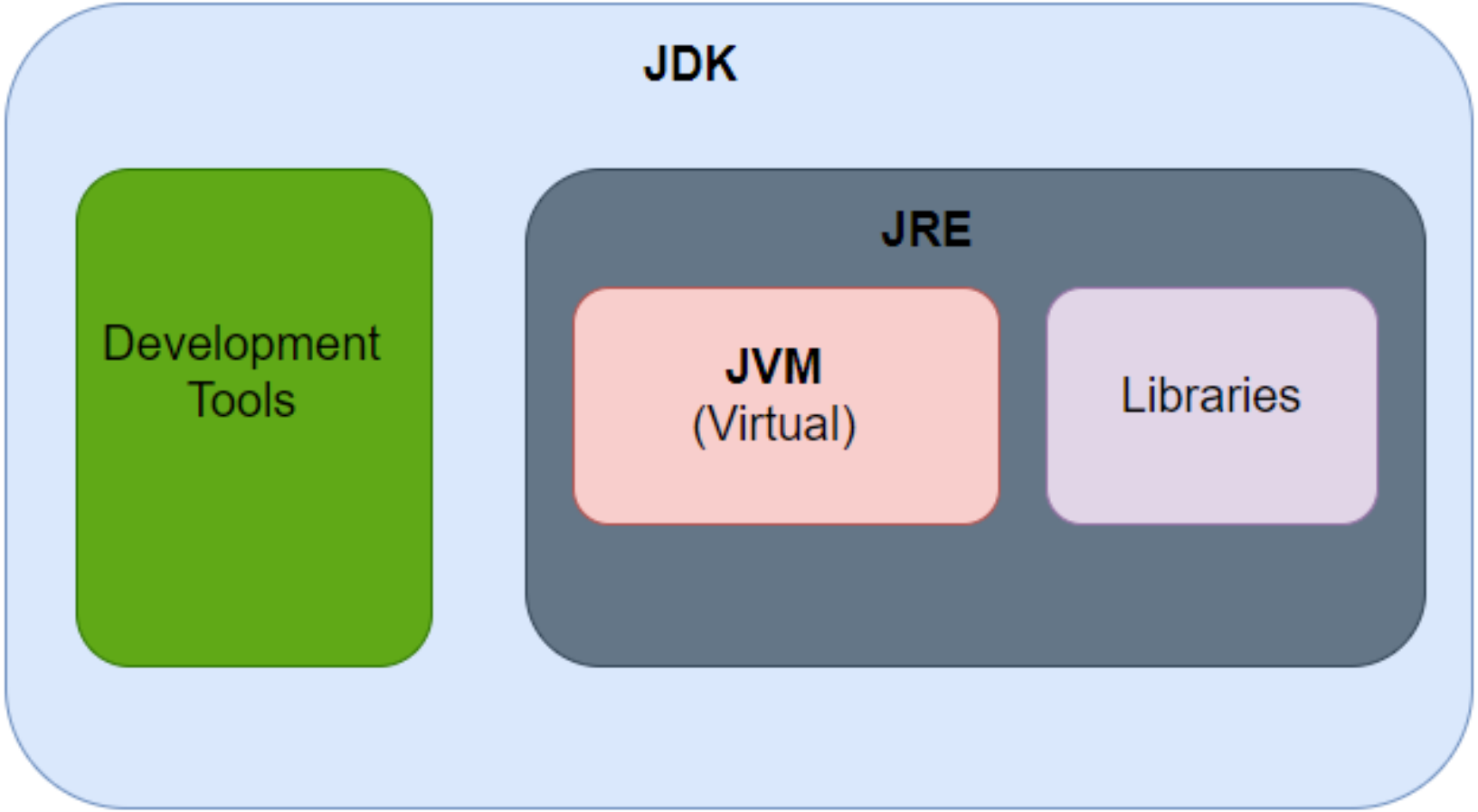




How Java Work?



How Java Work?



Java Install

- Some PCs might have Java already installed.

To check if you have Java installed on a Windows PC, search in the start bar for Java or type the following in Command Prompt (cmd.exe):

C:\Users*Your Name*>java -version

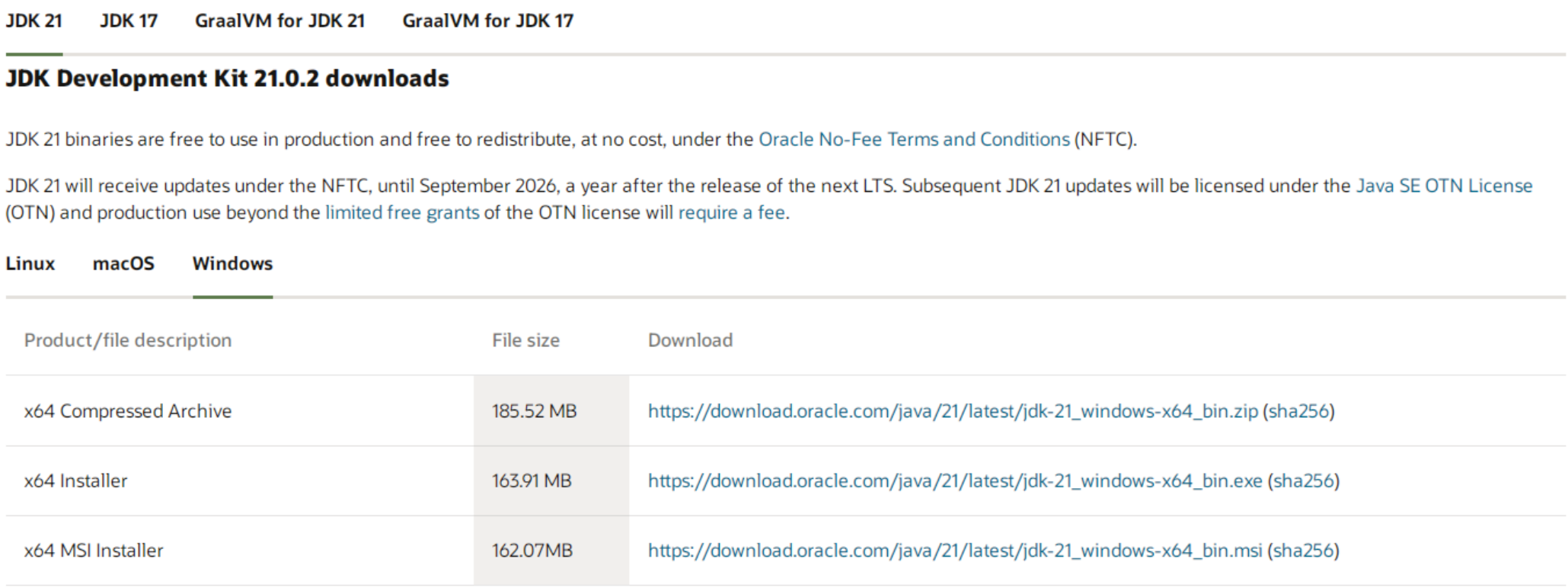
- If Java is installed, you will see something like this (depending on version):

java version "11.0.1" 2018-10-16 LTS
Java(TM) SE Runtime Environment 18.9 (build 11.0.1+13-LTS)
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.1+13-LTS, mixed mode)

- If you do not have Java installed on your computer, you can download it for free at oracle.com.

Java Install

Go to: https://www.oracle.com/java/technologies/downloads/#jdk21-windows



JDK 21 JDK 17 GraalVM for JDK 21 GraalVM for JDK 17

JDK Development Kit 21.0.2 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).

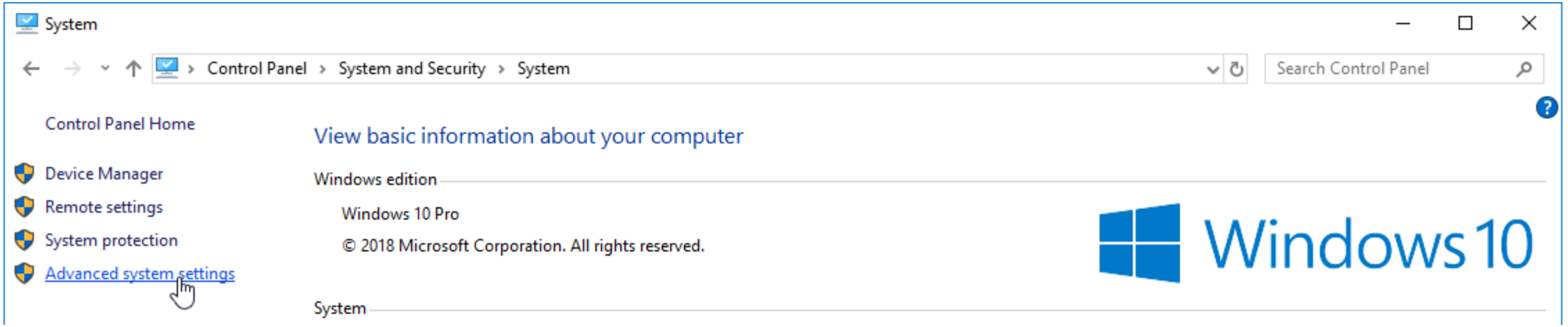
JDK 21 will receive updates under the NFTC, until September 2026, a year after the release of the next LTS. Subsequent JDK 21 updates will be licensed under the Java SE OTN License (OTN) and production use beyond the limited free grants of the OTN license will require a fee.

Linux macOS Windows

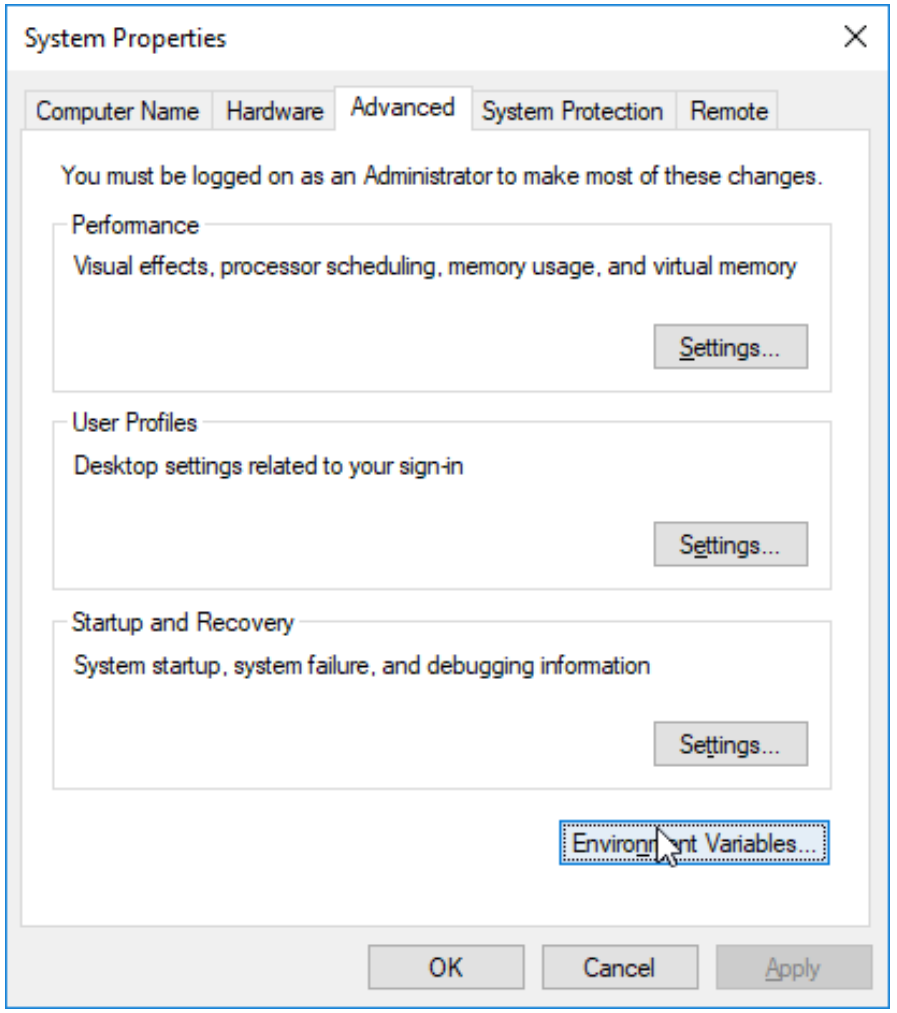
Product/file description	File size	Download
x64 Compressed Archive	185.52 MB	https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.zip (sha256)
x64 Installer	163.91 MB	https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.exe (sha256)
x64 MSI Installer	162.07MB	https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.msi (sha256)

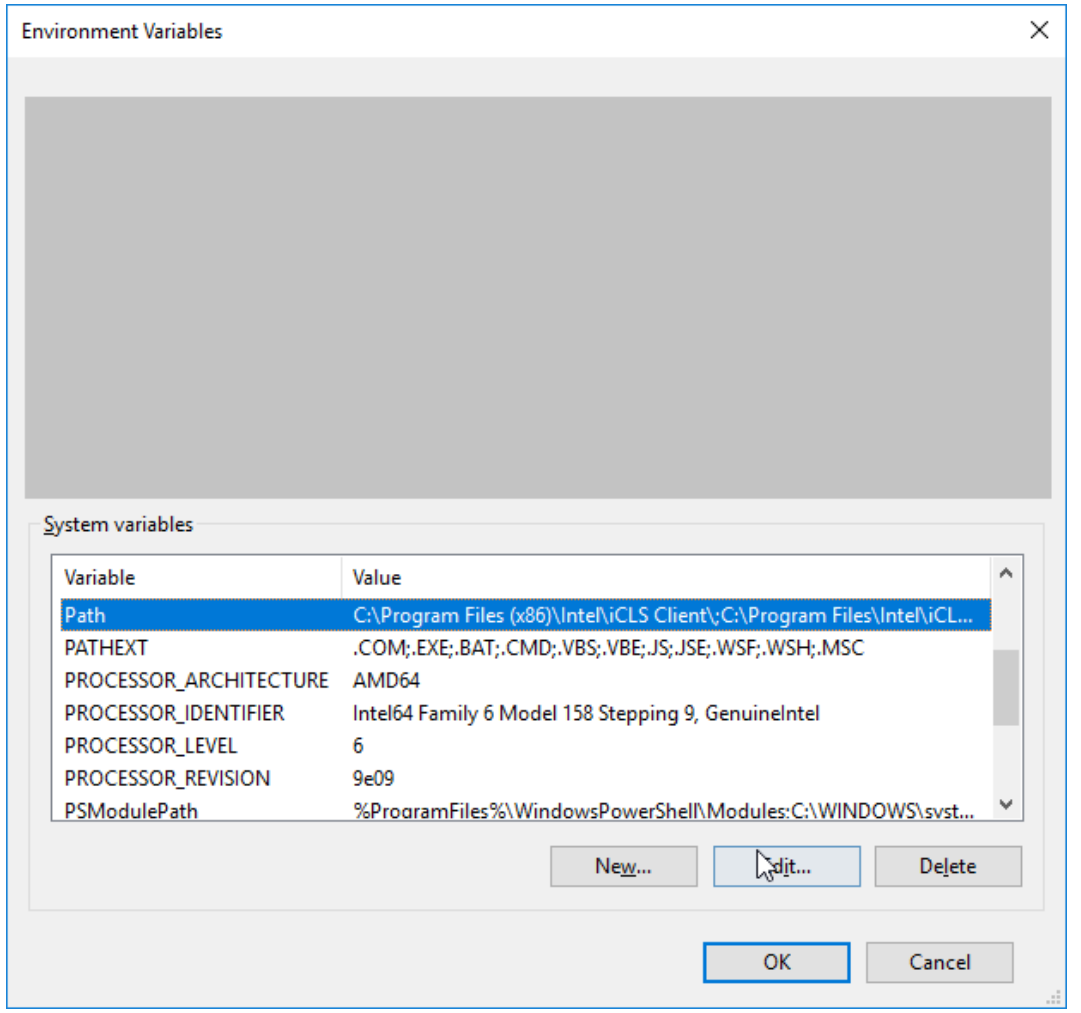
Setup for Windows

To install Java on Windows:

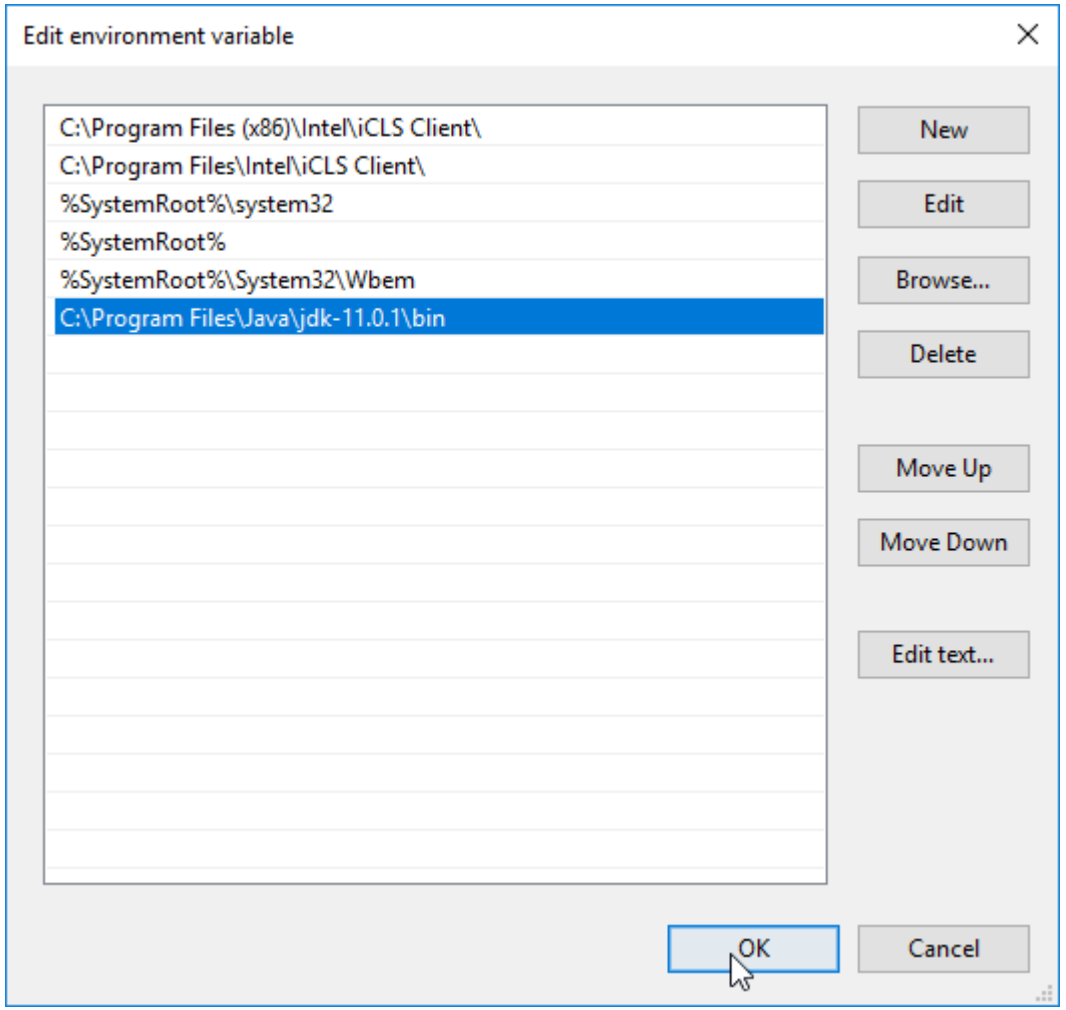


Setup for Windows





Setup for Windows



Write the following in the command line (cmd.exe):

C:\Users*Your Name*>java -version

If Java was successfully installed, you will see something like this (depending on version):

```
java version "11.0.1" 2018-10-16 LTS
Java(TM) SE Runtime Environment 18.9 (build
11.0.1+13-LTS)
Java HotSpot(TM) 64-Bit Server VM 18.9 (build
11.0.1+13-LTS, mixed mode)
```

Integrated Development Environment (IDE)

It is possible to write Java in an Integrated Development Environment, such as IntelliJ IDEA, Netbeans or Eclipse, which are particularly useful when managing larger collections of Java files.

Java Applications

In Java, every application begins with a class name, and that **class must match the filename**.

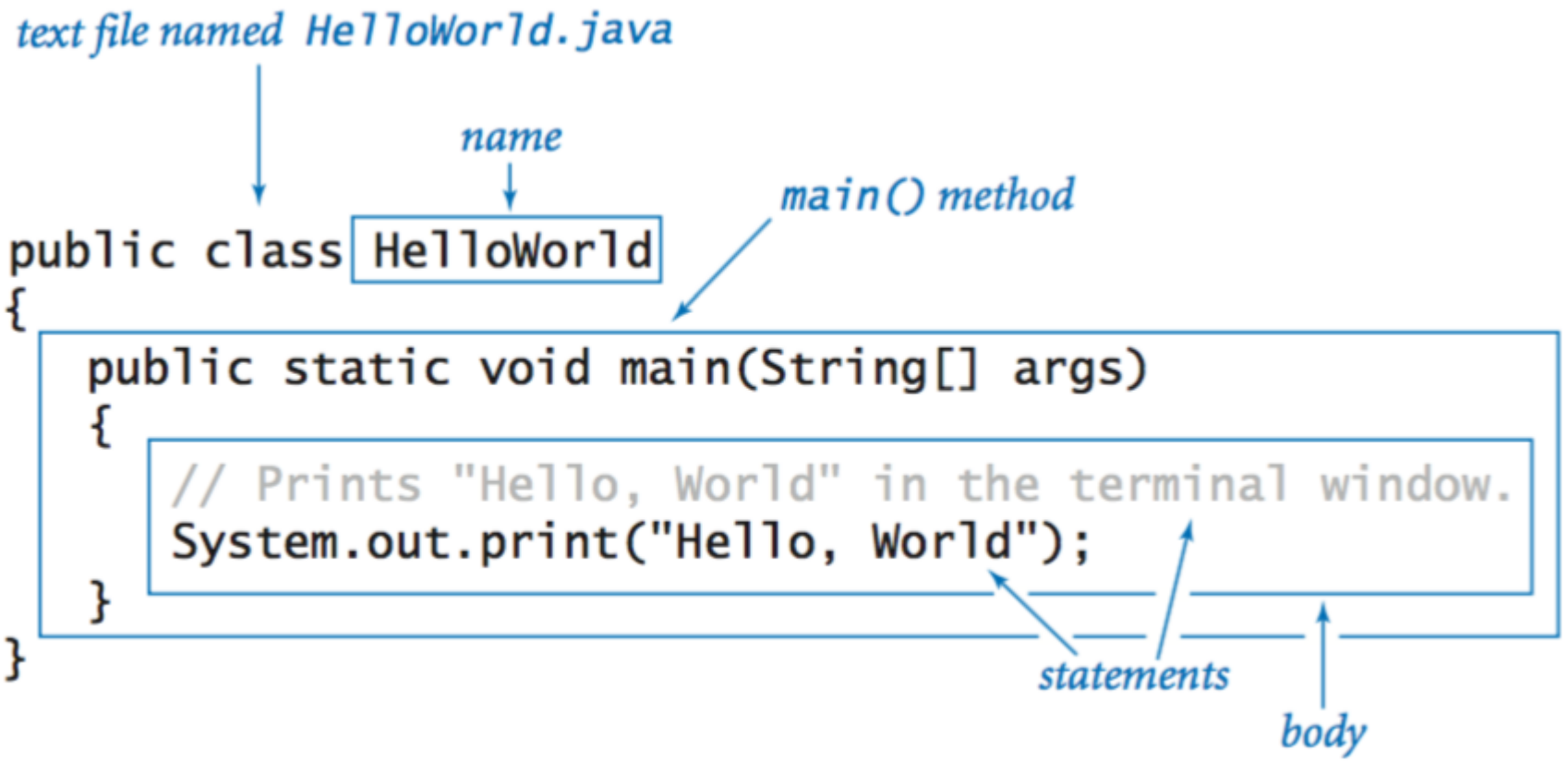
Let's create our first Java file, called Test.java, which can be done in any text editor (like Notepad).

The file should contain a "Hello World" message, which is written with the following code:

```
public class Test {

    public static void main(String[] args) {

        System.out.println("Hello World");

    }

}
```

```
C:\Users\Your Name>javac Test.java

C:\Users\Your Name>java Test
```

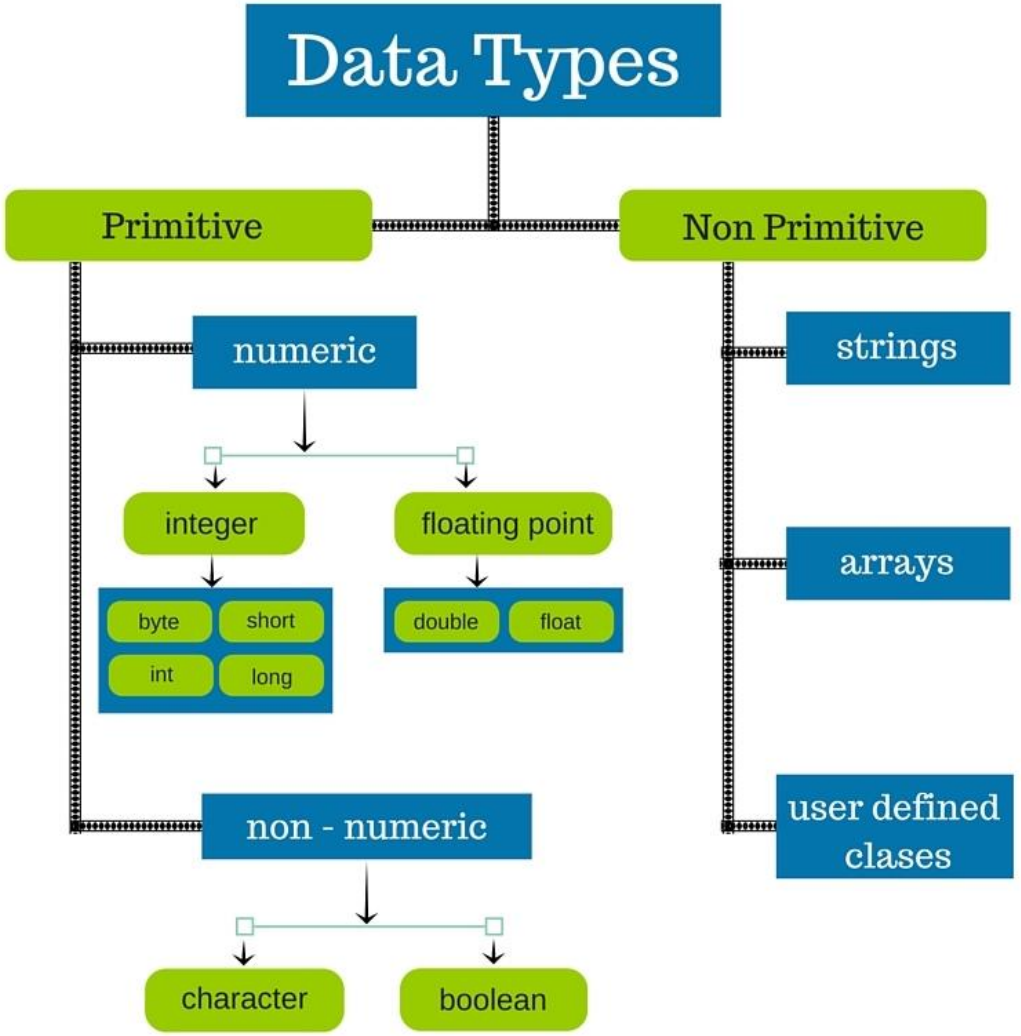
Java Syntax


text file named HelloWorld.java
name
main() method
public class HelloWorld
{
public static void main(String[] args)
{
// Prints "Hello, World" in the terminal window.
System.out.print("Hello, World");
}
}
statements
body

Java Comments

```
public class Test {

    public static void main(String[] args) {

        // This is a comment
        System.out.println("Ali");

        System.out.println("Ahmed"); // This is a comment

        /* The code below will print the words Hello World
        to the screen, and it is amazing */
        System.out.println("University");


    }
}
```

Data Types


Data Types
Primitive
Non Primitive
numeric
strings
integer
floating point
arrays
byte
short
int
long
double
float
non - numeric
user defined clases
character
boolean

Primitive Data Types

A primitive data type specifies the size and type of variable values, and it has no additional methods. There are eight primitive data types in Java:

Data Type	Size	Description
byte	1 byte	Stores whole numbers from -128 to 127
short	2 bytes	Stores whole numbers from -32,768 to 32,767
int	4 bytes	Stores whole numbers from -2,147,483,648 to 2,147,483,647
long	8 bytes	Stores whole numbers from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
float	4 bytes	Stores fractional numbers. Sufficient for storing 6 to 7 decimal digits
double	8 bytes	Stores fractional numbers. Sufficient for storing 15 decimal digits
boolean	1 bit	Stores true or false values
char	2 bytes	Stores a single character/letter or ASCII values

Java Variables

```
public class Test {
    public static void main(String[] args) {

        String name = "John";
        System.out.println(name);

        int myNum1 = 15;
        System.out.println(myNum1);

        final int myNum2 = 15;
        myNum2 = 20;  //will generate an error: cannot assign a value to a final variable
        System.out.println(myNum2);

    }
}
```

Final Variables: If you don't want others (or yourself) to overwrite existing values, use the final keyword (this will declare the variable as "final" or "constant", which means unchangeable and read-only).

Other Types

```
public class Test {
    public static void main(String[] args) {

        int myNum = 5;
        float myFloatNum = 5.99f;
        char myLetter = 'D';
        boolean myBool = true;
        String myText = "Hello";

        System.out.println(myNum);
        System.out.println(myFloatNum);
        System.out.println(myLetter);
        System.out.println(myBool);
        System.out.println(myText);

    }
}
```

Print Variables

```
public class Test {
    public static void main(String[] args) {

        String name = "John";
        System.out.println("Hello " + name);

        String firstName = "John ";
        String lastName = "Doe";
        String fullName = firstName + lastName;
        System.out.println(fullName);

        int x = 5;
        int y = 6;
        System.out.println(x + y); // Print the value of x + y

    }
}
```

Declare Multiple Variables

```
public class Test {
    public static void main(String[] args) {

        int x = 5, y = 6, z = 50;
        System.out.println(x + y + z);

        int x1, y1, z1;
        x1 = y1 = z1 = 50;
        System.out.println(x1 + y1 + z1);

    }
}
```

Identifiers

All Java variables must be identified with unique names. These unique names are called identifiers.

The general rules for naming variables are:

- Names can contain letters, digits, underscores (_), and dollar signs ($)
- Names must begin with a letter
- Names should start with a lowercase letter and it cannot contain whitespace
- Names can also begin with $ and _ (but we will not use it in this tutorial)
- Names are case sensitive ("myVar" and "myvar" are different variables)
- Reserved words (like Java keywords, such as int or boolean) cannot be used as names

```
public class Test {
    public static void main(String[] args)
     {         // Good
        int minutesPerHour = 60;
        System.out.println(minutesPerHour);

     }
}
```

Characters Data Type

The char data type is used to store a single character. The character must be surrounded by single quotes, like 'A' or 'c‘.

Alternatively, if you are familiar with ASCII values, you can use those to display certain characters:

```
public class Test {
    public static void main(String[] args) {

        char myGrade = 'B';
        System.out.println(myGrade);
        System.out.println("--------------");


        char myVar1 = 65, myVar2 = 66, myVar3 = 67;
        System.out.println(myVar1);
        System.out.println(myVar2);
        System.out.println(myVar3);

    }
}
```

```
B

--------------

A
B
C
```

Thanks

References: https://www.w3schools.com