Java Quick Start Overview

# Java Setup

## Installation

A guide for installing the latest Java JDK can be found here - <http://docs.oracle.com/javase/7/docs/webnotes/install/windows/jdk-installation-windows.html>

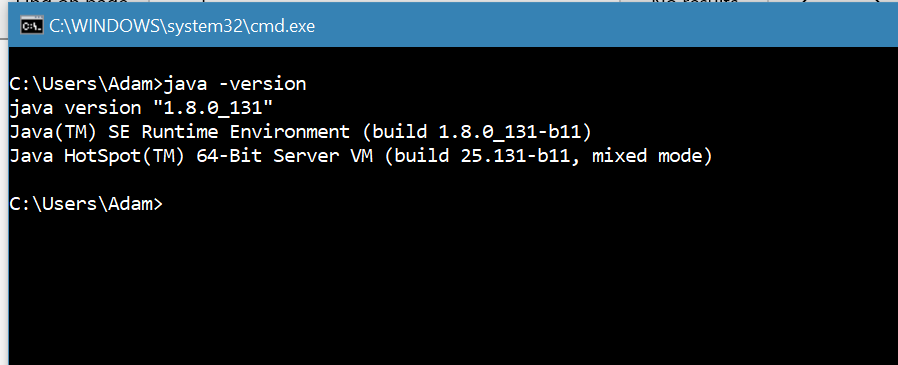
These instructions are written for Java SE 7, but they also apply to Java SE 8, just download the appropriate version

## Commands

Most Commands require you to have Java set on your System Path. Instructions are provided in the link from the previous section.

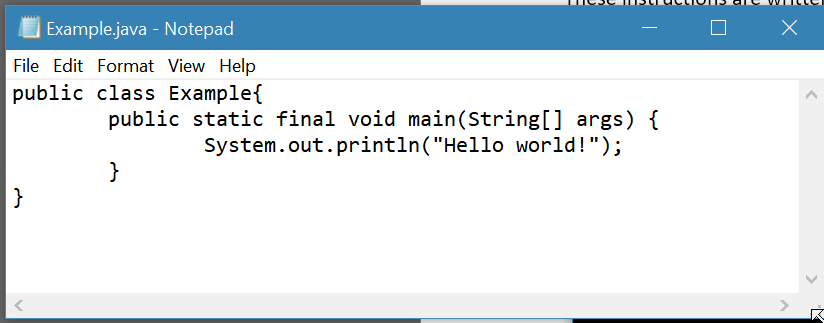
### Version

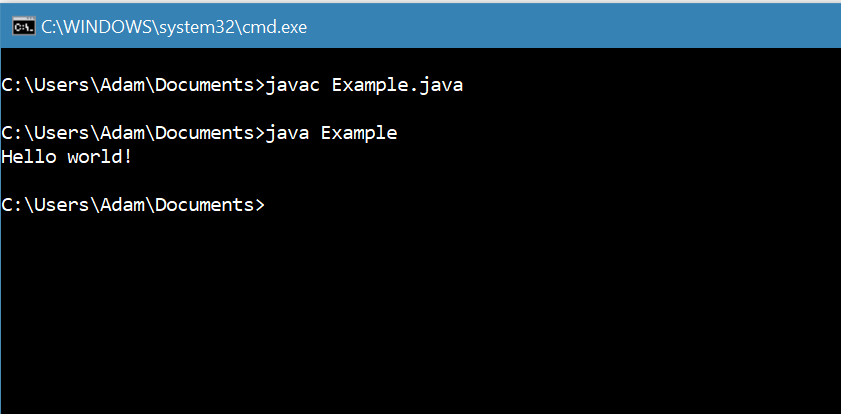
If Java is on your System Path, then you should be able to check the version via command line using the following command.



### javac

The javac command is used to compile .java files to .class files. This is not often used, as most projects use a build system like Maven. javac can be used for simple demos like the one shown below.





# Eclipse Setup

You can install Eclipse one of two ways. One will download, extract and create shortcuts for you, the other will just give you a .zip archive to extract. Here are links to.

### Installer

<https://www.eclipse.org/downloads/download.php?file=/oomph/epp/neon/R3/eclipse-inst-win64.exe>

Be sure to select the Java EE version, as it includes many plugins that help with Maven and web application development.

## Zip Archive

<http://www.eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/neon3>

# NetBeans Setup

NetBeans can be downloaded with the Java JDK as a bundle, or it can be downloaded and installed separately. You will need Admin rights on your PC to install.

<https://netbeans.org/downloads/>

The Java EE version is best for Web Application Development.

# Java Primitive Types and Operators

**Primitive Types**

|  |  |  |
| --- | --- | --- |
| **TYPE** | **DEFAULT VALUE** | **SIZE** |
| Boolean | False | 1 bit |
| Byte | 0 | 8 bits |
| Char | \u0000 | 16 bits |
| Short | 0 | 16 bits |
| Int | 0 | 32 bits |
| Long | 0 | 64 bits |
| Float | 0.0 | 32 bits |
| Double | 0.0 | 64 bits |

Java also has a number of built-in operators. These come in several types.

|  |  |  |
| --- | --- | --- |
| **Unary Operators** | **Example expression** | **Yields** |
| ++ | 1 ++ | 2 |
| -- | 1 -- | 0 |

|  |  |  |
| --- | --- | --- |
| **Arithmetic Operators** | **Example** | **Yields** |
| \* | 5 \* 4 | 20 |
| / | 4 / 2 | 2 |
| % | 27 % 5 | 2 |

|  |  |  |
| --- | --- | --- |
| **Shift Operators** | **Example** | **Yields** |
| << |  |  |
| >> |  |  |
| >>> |  |  |

|  |  |  |
| --- | --- | --- |
| **Relational Operators** | **Example** | **Yields** |
| < | 2 < 3 | True |
| > | 2 > 3 | False |
| <= | 3 <= 4 | True |
| >= | 3 >= 4 | True |
| instanceof | 1 instanceof String | False |

|  |  |  |
| --- | --- | --- |
| **Bitwise Operators** | **Example** | **Yields** |
| & |  |  |
| | |  |  |

|  |  |  |
| --- | --- | --- |
| **Logical Operators** | **Example** | **Yields** |
| && |  |  |
| || |  |  |

|  |  |  |
| --- | --- | --- |
| **Ternary Operators** | **Example** | **Yields** |
| ? |  |  |
| : |  |  |

|  |  |  |
| --- | --- | --- |
| **Assignment Operators** | **Example** | **Yields** |
| = |  |  |
| += |  |  |
| -= |  |  |
| \*= |  |  |
| /= |  |  |
| %= |  |  |
| &= |  |  |
| ^= |  |  |
| <<= |  |  |
| >>= |  |  |
| >>>= |  |  |

Operators in Yellow are not within the scope of this document and are not covered in any detail. See the Oracle Docs for more information – <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/operators.html>

# Java Abstract and Interface Types

Java is an Object Oriented Language[[1]](#footnote-1), meaning that all things are Objects and Object Attributes can be inherited from other Objects.

## Interface

An Interface can be thought of as a contract. All Classes that implement an interface must implement all the methods within that interface[[2]](#footnote-2). Here’s an Example Interface.

## Java Implements and Extends

## 

## @Override

# Java Entities – JPA and Hibernate

## @Entity

# Maven

## Installation

## Commands

## Using Maven with Netbeans

### Using Maven with Eclipse

# Getting the Demo Code and opening in your IDE

## Eclipse

## Netbeans

## Generic Text Editor (advanced)

# Running the Demo Code

## Eclipse

## Netbeans

## Generic Text Editor

1. There is some debate within the developer community as to the ‘purity’ of Java’s Object Orientation, mostly because of the primitive types not inheriting from *Object*. [↑](#footnote-ref-1)
2. An Abstract Class Implementing an Interface does not have to implement all the methods, since an Abstract Class is not considered to be “Concrete” [↑](#footnote-ref-2)