

THE EXPERT'S VOICE® IN JAVA

Java Quick Syntax Reference

Mikael Olsson

Apress®

For your convenience Apress has placed some of the front matter material after the index. Please use the Bookmarks and Contents at a Glance links to access them.



Apress®

Contents at a Glance

About the Author	xi
Introduction	xiii
■ Chapter 1: Hello World.....	1
■ Chapter 2: Compile and Run	3
■ Chapter 3: Variables	5
■ Chapter 4: Operators	9
■ Chapter 5: String	13
■ Chapter 6: Arrays	15
■ Chapter 7: Conditionals	19
■ Chapter 8: Loops.....	21
■ Chapter 9: Methods	25
■ Chapter 10: Class	29
■ Chapter 11: Static.....	35
■ Chapter 12: Inheritance.....	39
■ Chapter 13: Overriding	41
■ Chapter 14: Packages and Import	45
■ Chapter 15: Access Levels.....	47
■ Chapter 16: Constants	51
■ Chapter 17: Interface.....	53

■ **Chapter 18: Abstract** **57**

■ **Chapter 19: Enum** **59**

■ **Chapter 20: Exception Handling** **61**

■ **Chapter 21: Boxing and Unboxing** **65**

■ **Chapter 22: Generics** **67**

Index..... **73**

Introduction

Java is a high-level object-oriented programming language developed by Sun Microsystems, which became part of Oracle Corporation in 2010. The language is very similar to C++, but has been simplified to make it easier to write bug free code. Most notably, there are no pointers in Java, instead all memory allocation and deallocation is handled automatically.

Despite simplifications like this Java has considerably more functionality than both C and C++, due to its large class library. Java programs also have high performance and can be made very secure, which has contributed to making Java the most popular general purpose programming language in use today.

Another key feature of Java is that it is platform independent. This is achieved by only compiling programs half-way, into platform independent instructions called bytecode. The bytecode is then interpreted, or run, by the Java Virtual Machine (JVM). This means that any system that has this program and its accompanying libraries installed can run Java applications.

There are three class libraries available for the Java programming language: Java ME, Java SE and Java EE. Java ME (Mobile Edition) is a stripped down version of Java SE (Standard Edition), while Java EE (Enterprise Edition) is an extended version of Java SE that includes libraries for building web applications.

The Java language and class libraries have undergone major changes since their initial release in 1996. The naming conventions for the versions have gone through a few revisions as well. The major releases include: JDK 1.0, JDK 1.1, J2SE 1.2, J2SE 1.3, J2SE 1.4, J2SE 5.0, Java SE 6 and Java SE 7, which is the current version as of writing.

After J2SE 1.4 the version number was changed from 1.5 to 5.0 for marketing reasons. As of J2SE 5.0, there is one version number for the product and another one used internally by the developers. J2SE 5.0 is the product name, while Java 1.5 is the developer version. Similarly, Java SE 7 is the product and Java 1.7 the internal version number. For simplicity's sake, the Java versions will be referred to as Java 1-7 in this book. Note that Java is designed to be backwards compatible. Thus the Virtual Machine for Java 7 can still run Java 1 class files.

CHAPTER 1



Hello World

Installing

Before you can program in Java you need to download and install the Java Development Kit (JDK) Standard Edition (SE) from Oracle's website.¹ Among other things, the JDK includes the Java compiler, the class libraries and the virtual machine needed to run Java applications. Oracle's download page also has a link to obtain Netbeans² bundled with the JDK. Netbeans is an Integrated Development Environment (IDE) that will make development in Java much easier. Alternatively, another free IDE you can use is Eclipse,³ or if you do not want to use any IDE at all a regular text editor will work just fine.

Creating a project

If you decide to use an IDE (recommended) you need to create a project, which will manage the Java source files and other resources. Alternatively, if you prefer not to use an IDE you can create an empty file with the .java extension, for example MyApp.java, and open it in your text editor of choice.

To create a project in Netbeans, go to the File menu and select New Project. From the dialog box select the Java Application project type under the Java category and click next. On this dialog box set the project name to "MyProject" and the name of the main class to "myproject.MyApp". Change the project's location if you want to, and then hit the Finish button to generate the project. The project's only file, MyApp.java, will then open up, containing some default code. You can go ahead and remove all of that code so that you start with an empty source file.

Hello world

When you have your project and programming environment set up the first application you will create is the Hello World program. This program will teach you how to compile and run Java applications, as well as how to output a string to a command window.

¹<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

²<http://www.netbeans.org>

³<http://www.eclipse.org>

The first step in creating this program is to add a public class to your `MyApp.java` source file. The class must have the same name as the physical source file without the file extension, in this case “`MyApp`”. It is legal to have more than one class per file in Java, but only one public class is allowed, and that name must match the filename. Keep in mind that Java is case sensitive. The curly brackets following the class name delimits what belongs to the class and must be included. The brackets, along with their content, is referred to as a code block, or just a block.

```
public class MyApp {}
```

Next, add the main method inside the class. This is the starting point of the application and must always be included in the same form as is shown below. The keywords themselves will be looked at in later chapters.

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

The last step in completing the Hello World program is to output the text by calling the `print` method. This method is located inside the built-in `System` class, and then another level down inside the `out` class. The method takes a single argument – the string to be printed – and it ends with a semicolon, as do all statements in Java.

```
public class MyApp {
    public static void main(String[] args) {
        System.out.print("Hello World");
    }
}
```

Note that the dot operator (`.`) is used to access members of a class.

Code hints

If you are unsure of what a specific class contains, or what arguments a method takes, you can take advantage of code hints in some IDEs, such as Netbeans. The code hint window appears anytime you are typing code and there are multiple predetermined alternatives. It can also be brought up manually by pressing `Ctrl + Space`. This is a very powerful feature that gives you quick access to the whole class library and their members, along with descriptions.

CHAPTER 2



Compile and Run

Running from the IDE

With your Hello World program complete you can compile and run it in one of two ways. The first method is by selecting run from the menu bar of the IDE that you are using. In Netbeans the menu command is: Run ► Run Main Project. The IDE will then compile and run the application, which displays the text “Hello World”.

Running from a console window

The other way is to manually compile the program by using a console window (C:\Windows\System32\cmd.exe). The most convenient way to do this is to first add the JDK bin directory to the PATH environment variable. In Windows, this can be done by using the SET PATH command, and then by appending the path to your JDK installation’s bin folder separated by a semicolon.

```
SET PATH=%PATH%;"C:\Program Files\JDK\bin"
```

By doing this the console will be able to find the Java compiler from any folder for the duration of this console session. The PATH variable can also be permanently changed.¹ Next, navigate to the folder where the source file is located and run the compiler by typing “javac” followed by the complete filename.

```
javac MyApp.java
```

The program will be compiled into a class file called MyApp.class. This class file contains bytecode instead of machine code, so to execute it you need to call the Java Virtual Machine by typing “java” followed by the filename.

```
java MyApp
```

Notice that the .java extension is used when compiling a file, but the .class extension is not used when running it.

¹<http://www.java.com/en/download/help/path.xml>

Comments

Comments are used to insert notes into the source code and will have no effect on the end program. Java has the standard C++ comment notation, with both single-line and multi-line comments.

```
// single-line comment
```

```
/* multi-line  
   comment */
```

In addition to these, there is the Javadoc comment. This comment is used to generate documentation by using a utility included in the JDK bin folder which is also called Javadoc.

```
/** javadoc  
    comment */
```

CHAPTER 3



Variables

Variables are used for storing data during program execution.

Data types

Depending on what data you need to store there are several kinds of data types. Java has eight types that are built into the language. These are called *primitives*. The integer (whole number) types are byte, short, int and long. The float and double types represent floating-point numbers (real numbers). The char type holds a Unicode character and the boolean type contains either a true or false value. Except for these primitive types, every other type in Java is represented by either a class, an interface or an array.

Data Type	Size (bits)	Description
byte	8	Signed integer
short	16	
int	32	
long	64	
float	32	Floating-point number
double	64	
char	16	Unicode character
boolean	1	Boolean value

Declaring variables

To declare (create) a variable you start with the data type you want it to hold followed by a variable name. The name can be anything you want, but it is a good idea to give your variables names that are closely related to the values they will hold. The standard naming