

## Introduction

### About JAVA

“Java refers to a number of computer software products and specifications from Sun Microsystems (the Java™ technology) that together provide a system for developing and deploying cross-platform applications. Java is used in a wide variety of computing platforms spanning from embedded devices and mobile phones on the low end to enterprise servers and super computers on the high end. Java is fairly ubiquitous in mobile phones, Web servers and enterprise applications, and somewhat less common in desktop applications, though users may have come across Java applets when browsing the Web.

Writing in the Java programming language is the primary way to produce code that will be deployed as Java bytecode, though there are compilers available for other languages such as JavaScript, Python and Ruby, and a native Java scripting language called Groovy. Java syntax borrows heavily from C and C++ but it eliminates certain low-level constructs such as pointers and has a very simple memory model where every object is allocated on the heap and all variables of object types are references. Memory management is handled through integrated automatic garbage collection performed by the Java Virtual Machine (JVM).”<sup>1</sup>

### OOP – Object Oriented Programming

OOP is a particular style of programming which involves a particular way of designing solutions to particular problems. Most modern programming languages, including Java, support this paradigm. When speaking about OOP one has to mention:

- Inheritance
- Modularity
- Polymorphism
- Encapsulation (binding code and its data)

However at this point it is too early to try to fully understand these concepts.

This guide is divided into two major sections, the first section is an introduction to the language and illustrates various examples of code while the second part goes into more detail.

---

<sup>1</sup> [http://en.wikipedia.org/wiki/Java\\_%28Sun%29](http://en.wikipedia.org/wiki/Java_%28Sun%29)

## Part 1 - Getting Started

### The Java Development Kit - JDK

In order to get started in Java programming, one needs to get a recent copy of the Java JDK. This can be obtained for free by downloading it from the Sun Microsystems website, <http://java.sun.com/>

Once you download and install this JDK you are ready to get started. You need a text editor as well and Microsoft's Notepad (standard with all Windows versions) suits fine.

### My first Java program

Open your text editor and type the following lines of code:

```
/*
My first program
Version 1
*/

public class Example1 {
    public static void main (String args []) {
        System.out.println ("My first Java program");
    }
}
```

This is known as a Block Comment.  
These lines are useful to the  
programmer and are ignored by the  
Compiler

Save the file as Example1.java<sup>2</sup>. The name of the program has to be similar to the filename. Programs are called classes. Please note that Java is **case-sensitive**. You cannot name a file "Example.java" and then in the program you write "public class example". It is good practice to insert comments at the start of a program to help you as a programmer understand quickly what the particular program is all about. This is done by typing "/\*" at the start of the comment and "\*/" when you finish. The predicted output of this program is:

```
My first Java program
```

In order to get the above output we have to first compile the program and then execute the compiled class. The applications required for this job are available as part of the JDK:

- javac.exe – compiles the program
- java.exe – the interpreter used to execute the compiled program

In order to compile and execute the program we need to switch to the command prompt. On windows systems this can be done by clicking Start>Run>cmd

---

<sup>2</sup> Ideally you should create a folder on the root disk (c:\) and save the file there

At this point one needs some basic DOS commands in order to get to the directory (folder), where the java class resides:

- `cd\` (change directory)
- `cd\[folder name]` to get to the required folder/directory

When you get to the required destination you need to type the following:

```
c:\[folder name]\javac Example1.java
```

The above command will compile the java file and prompt the user with any errors. If the compilation is successful a new file containing the bytecode is generated: **Example1.class**

To execute the program, we invoke the interpreter by typing:

```
c:\[folder name]\java Example1
```

The result will be displayed in the DOS window.

### Using an IDE

Some of you might already be frustrated by this point. However there is still hope as one can forget about the command prompt and use an IDE (integrated development environment) to work with Java programming. There are a number of IDE's present, all of them are fine but perhaps some are easier to work with than others. It depends on the user's level of programming and tastes! The following is a list of some of the IDE's available:

- BlueJ – [www.bluej.org](http://www.bluej.org) (freeware)
- NetBeans – [www.netbeans.org](http://www.netbeans.org) (freeware/open-source)
- JCreator – [www.jcreator.com](http://www.jcreator.com) (freeware version available, pro version purchase required)
- Eclipse – [www.eclipse.org](http://www.eclipse.org) (freeware/open-source)
- IntelliJ IDEA – [www.jetbrains.com](http://www.jetbrains.com) (trial/purchase required)
- JBuilder – [www.borland.com](http://www.borland.com) (trial/purchase required)

Beginners might enjoy BlueJ and then move onto other IDE's like JCreator, NetBeans, etc. Again it's just a matter of the user's tastes and software development area.