Understanding OOP and Java

Written by Paul Pu
www.torontocollege.com

About the Java Technology

Java technology is both a programming language and a platform.

The Java Programming Language

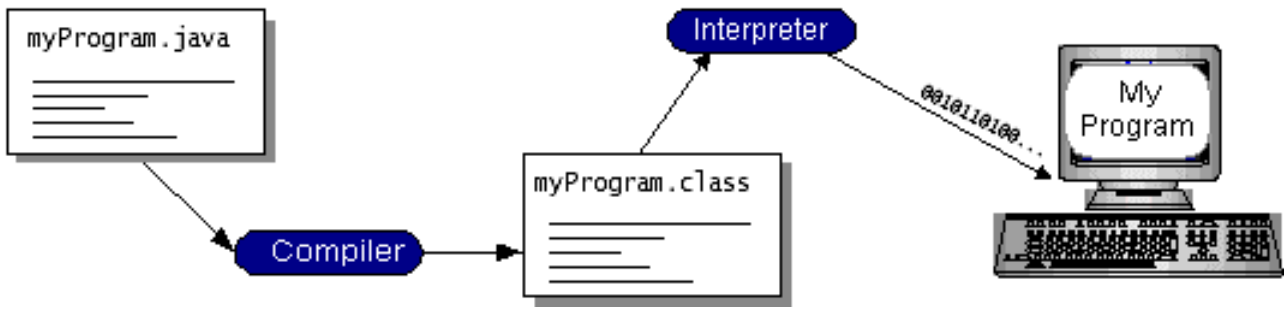
The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

The Java Programming Language

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

The Java Programming Language

- Java is both compiled and interpreted.



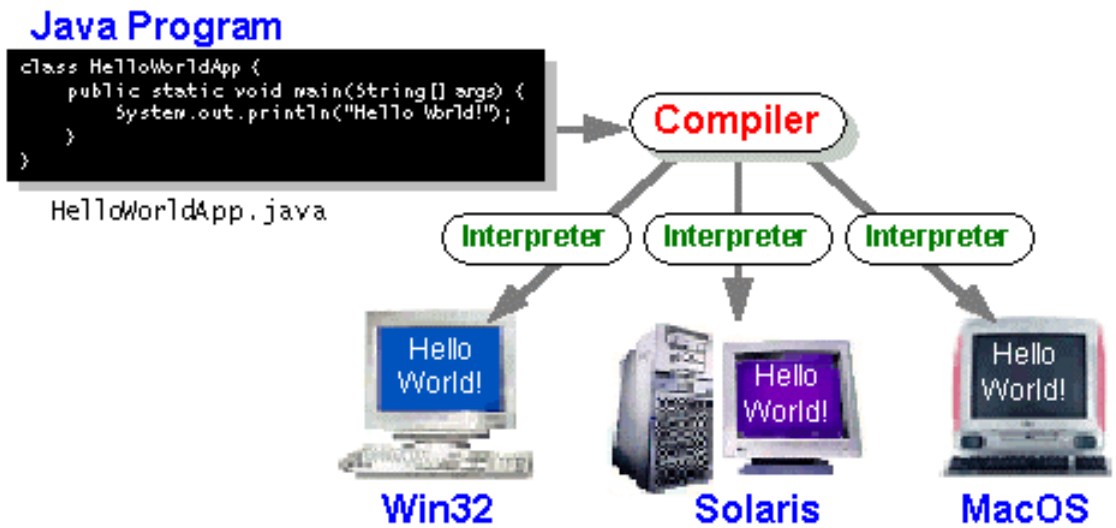
The Java Programming Language

- ***Java bytecodes*** — the platform-independent codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java bytecode instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The following figure illustrates how this works.
- Java bytecodes help make "write once, run anywhere" possible.

The Java Programming Language

- *Java Virtual Machine* :

software environment in which Java **bytecodes** run



The The Java Platform

- A *platform* is the hardware or software environment in which a program runs. We've already mentioned some of the most popular platforms like Windows 2000, Linux, Solaris, and MacOS. Most platforms can be described as a combination of the operating system and hardware. ***The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms.***

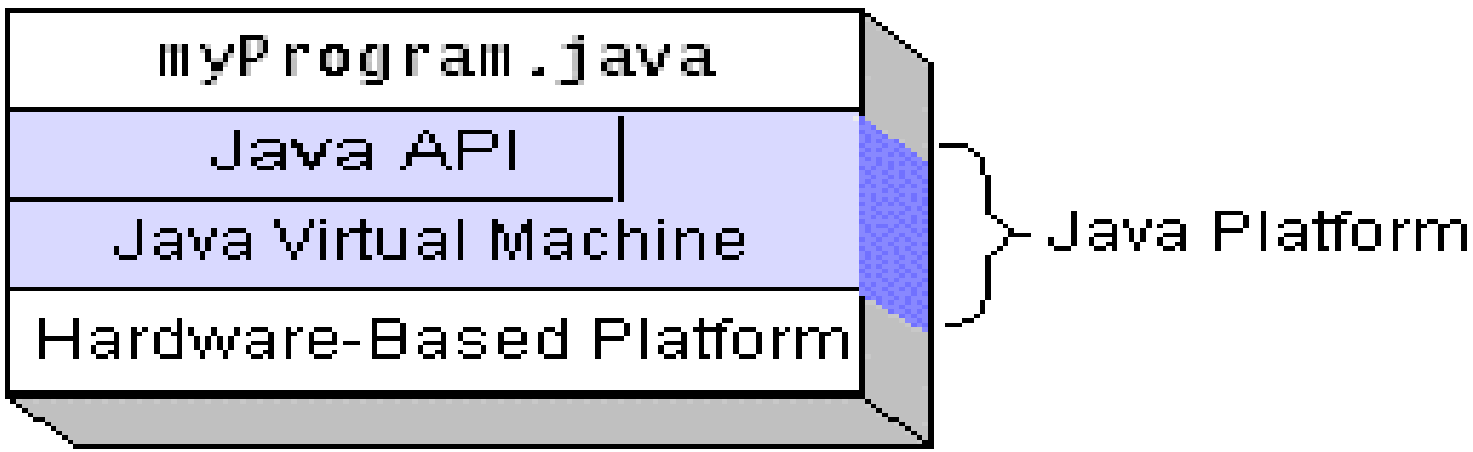
The The Java Platform

The Java platform has two components:

- The *Java Virtual Machine* (Java VM)
- The *Java Application Programming Interface* (Java API)

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets.

A program that's running on the Java platform



Written by Paul Pu
www.torontocollege.com

A program that's running on the Java platform

Native code is code that after you compile it, the compiled code runs on a specific hardware platform. As a platform-independent environment, the Java platform can be a bit slower than native code. However, smart compilers, well-tuned interpreters, and just-in-time bytecode compilers can bring performance close to that of native code without threatening portability.

What Can Java Technology Do?

The most common types of programs written in the Java programming language are *applets* and *applications*.

What Can Java Technology Do?

Every full implementation of the Java platform gives you the following features:

The essentials: Objects, strings, threads, numbers, input and output, data structures, system properties, date and time, and so on.

Applets: The set of conventions used by applets.

Internationalization: Help for writing programs that can be localized for users worldwide. Programs can automatically adapt to specific locales and be displayed in the appropriate language.

What Can Java Technology Do?

Every full implementation of the Java platform gives you the following features:

- **Networking**: URLs, TCP (Transmission Control Protocol), UDP (User Datagram Protocol) sockets, and IP (Internet Protocol) addresses.
- **Security**: Both low level and high level, including electronic signatures, public and private key management, access control, and certificates.
- **Software components**: Known as JavaBeans, can plug into existing component architectures.
- **Object serialization**: Allows lightweight persistence and communication via Remote Method Invocation (RMI).

What Can Java Technology Do?

Every full implementation of the Java platform gives you the following features

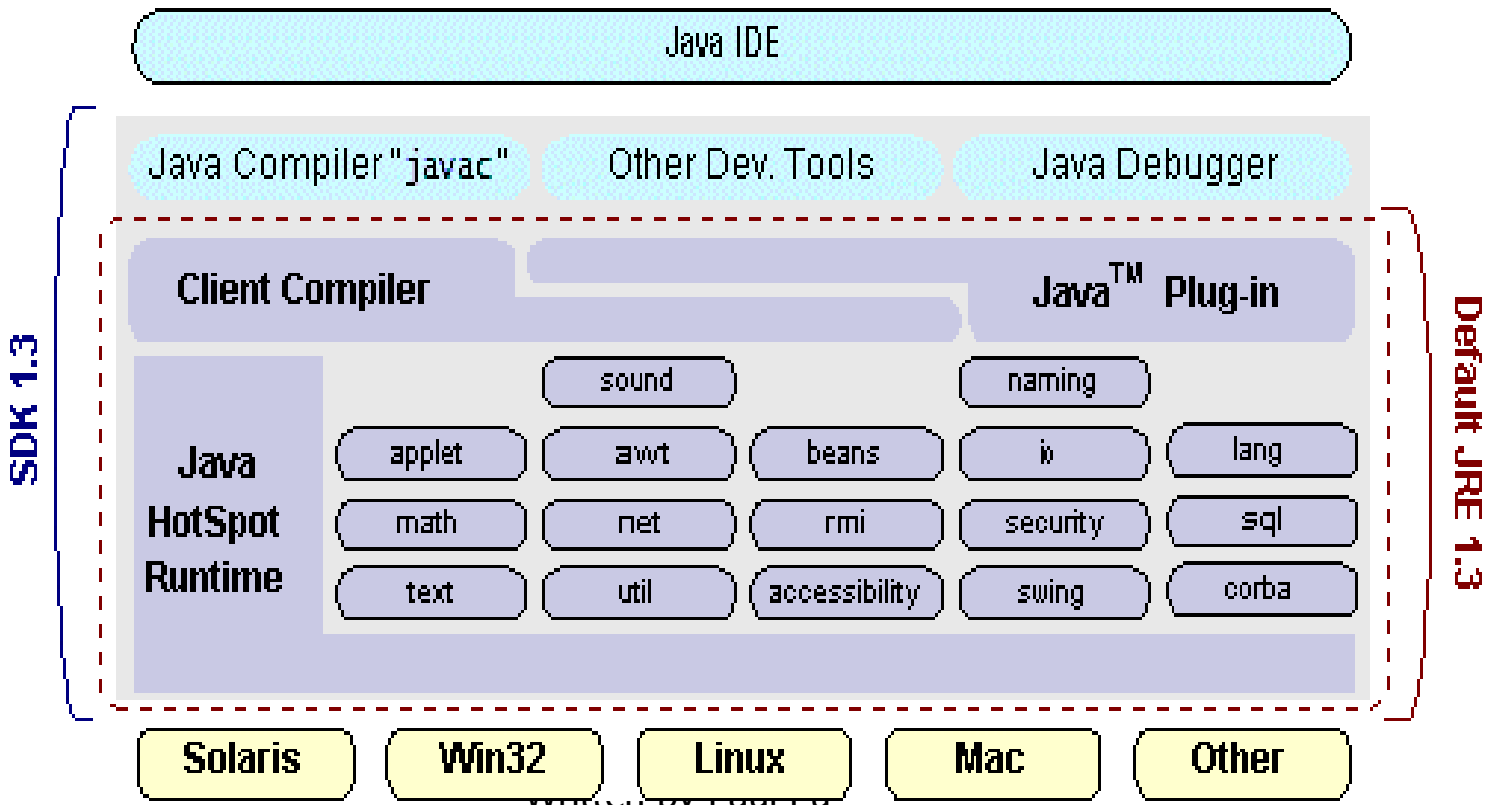
Java Database Connectivity (JDBC™): Provides uniform access to a wide range of relational databases.

Servlet and JSP

2D and 3D graphics, accessibility, servers, collaboration, telephony, speech, animation, and more.

What Can Java Technology Do?

Java 2 SDK.



How Will Java Technology Change My Life?

- Get started quickly
- Write less code
- Write better code
- Develop programs more quickly
- Avoid platform dependencies with 100% Pure Java
- Write once, run anywhere
- Distribute software more easily

Getting Started
First Java Application:HelloWord

```
public class HelloWorld {
  public static void main(String[] args) {
    System.out.println("HelloWorld! This is my First Java Program");
  }
}
```

Getting Started
First Java Applet:HelloWord

```
import java.applet.Applet;
import java.awt.Graphics;
public class HelloWorld extends Applet {
  public void paint(Graphics g) {
    g.drawString("Hello world!", 50, 25);
  }
}
```

Getting Started
The Anatomy of an Applet

- **Importing Classes and Packages**

```
import  java.applet.Applet;
import  java.awt.Graphics;
```

- ***Defining an Applet Subclass***

```
public class HelloWorld extends Applet {
```

Written by Paul Pu
www.torontocollege.com

Getting Started
The Anatomy of an Applet

- **Implementing Applet Methods**

```
public class HelloWorld extends Applet {
  public void paint(Graphics g) {
  g.drawString("Hello world!", 50, 25);
  }
}
```

The paint() method comes from Applet.

Getting Started
The Anatomy of an Applet

- **Running an Applet**
- <HTML>
- <HEAD>
- <TITLE> A Simple Program </TITLE>
- </HEAD>
- <BODY> Here is the output of my program:
- **<APPLET CODE="HelloWorld.class" WIDTH=150 HEIGHT=25>**
- **</APPLET>**
- </BODY>
- </HTML>

Getting Started

javac: **Command not found** . Check and make sure you do have JDK installed. On unix, check **setenv.**

- testing.java:14: `;' expected. System.out.println("Input has " + count + " chars.") ^ 1 error
- In class *classname*: void main(String argv[]) is not defined

Getting Started Set path and classpath

- # set path=%path%;/jdk1.3/bin
- # set classpath=%classpath%;.
- You can set path and classpath in:
- 1) DOS terminal
- 2)autoexec.bat file(for win95,win98)
- 3)environment variable(for winNT,win2000)
- 4)Shell file(for Unix/Lunix)

Microsoft .NET Framework

Written by Paul Pu
www.torontocollege.com

.NET and Java

.NET Framework	⟷	Java Platform
CLR	⟷	JVM
MSIL	⟷	Byte Codes
FCL	⟷	Java API

Written by Paul Pu
www.torontocollege.com

C# program

```
using System;

Class HelloWorld {
 static void Main(string[] args){
   Console.WriteLine("HelloWorld,C#");
}

}
```

C# program

```
using System;
using System.Windows.Forms;

Class Welcome {
static void Main(string[] args) {
MessageBox.Show("Welcome\nto\nC#\nprogramming!");
}
}
```