Getting Started with Java

Atul Prakash

Running Programs

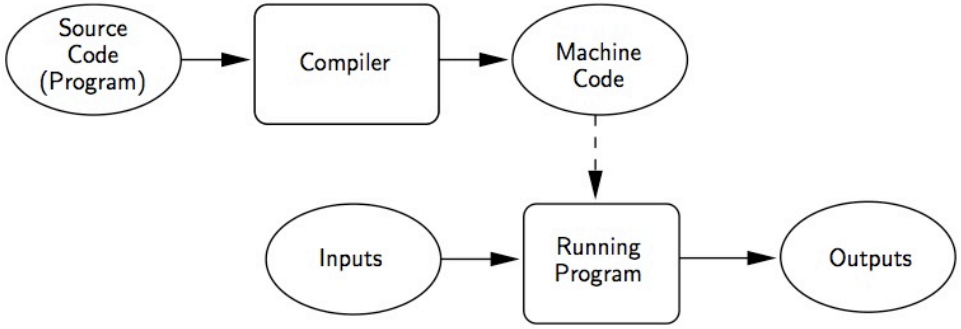


Figure 1.2: Compiling a High-Level Language

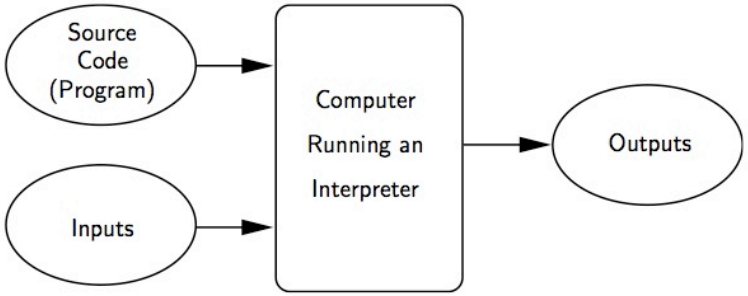
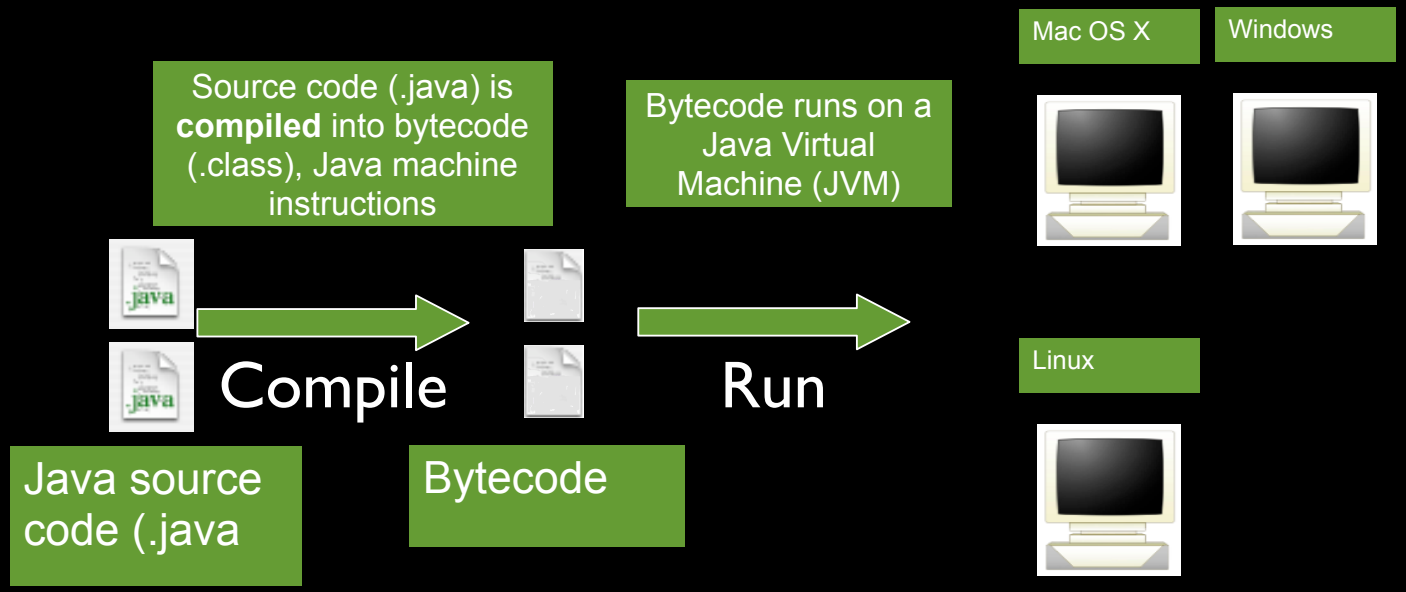


Figure 1.3: Interpreting a High-Level Language.

C++, Fortran, Pascal

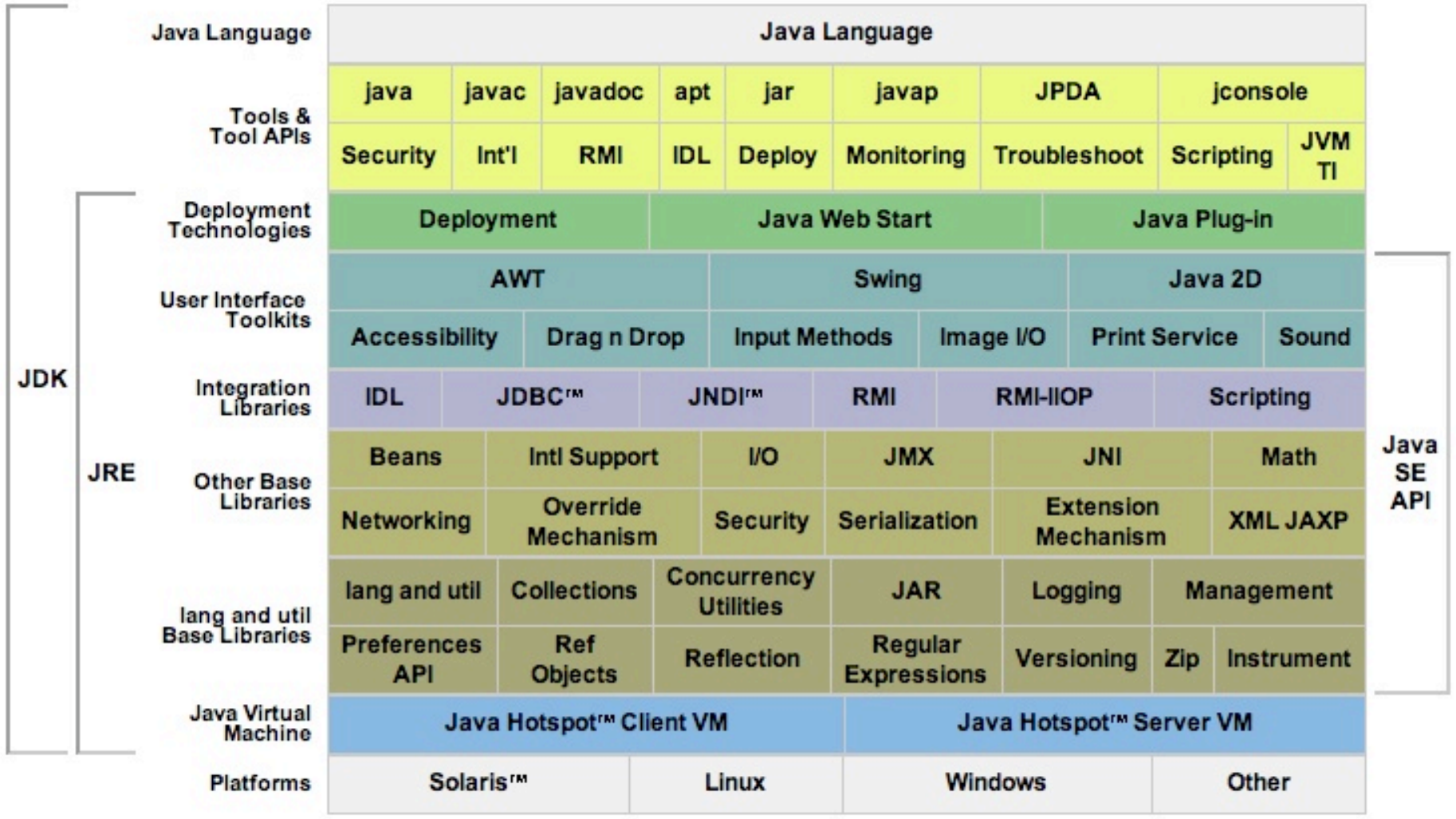
Python, PHP, Ruby, Perl

Java is compiled into device-independent code and then interpreted



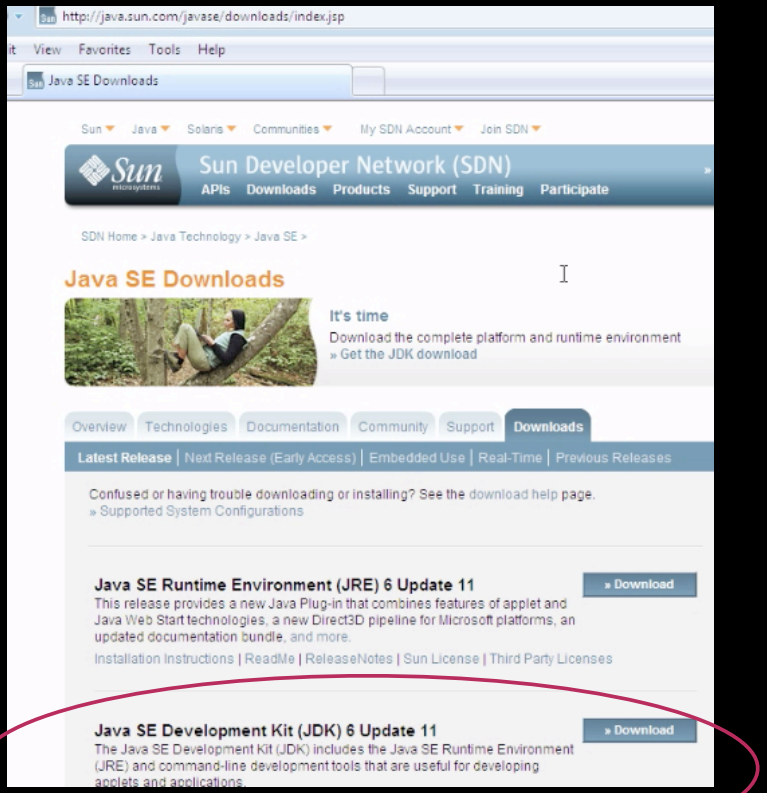
Java Overview: JRE & JDK

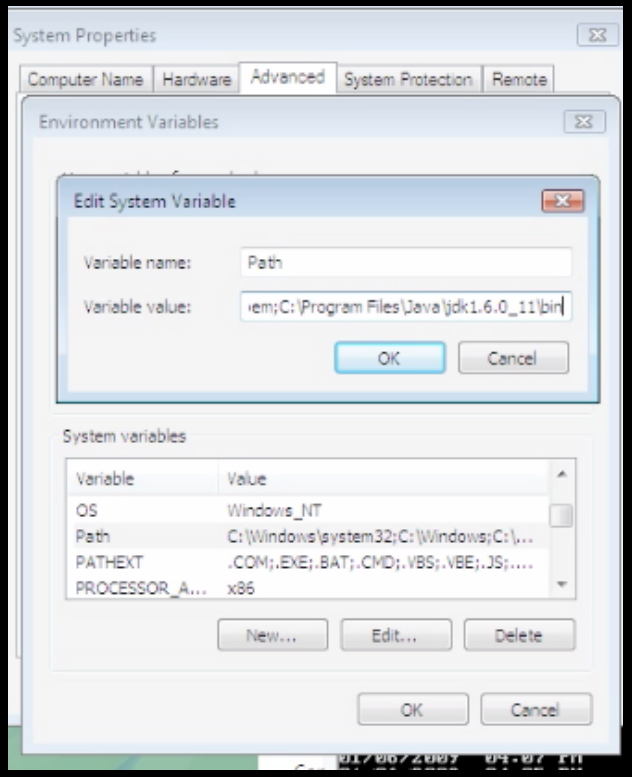
- There are two main software products in the Java Platform Standard Edition (Java SE):
  - Java SE Runtime Environment (JRE)
    - No command-line tools
  - Java SE Development Kit (JDK)
    - JRE + command-line tools


JDK
JRE
Java Language
Java Language
Tools & Tool APIs
java
javac
javadoc
apt
jar
javap
JPDA
jconsole
Security
Int'l
RMI
IDL
Deploy
Monitoring
Troubleshoot
Scripting
JVM TI
Deployment Technologies
Deployment
Java Web Start
Java Plug-in
User Interface Toolkits
AWT
Swing
Java 2D
Accessibility
Drag n Drop
Input Methods
Image I/O
Print Service
Sound
Integration Libraries
IDL
JDBC™
JNDI™
RMI
RMI-IIOP
Scripting
Other Base Libraries
Beans
Intl Support
I/O
JMX
JNI
Math
Networking
Override Mechanism
Security
Serialization
Extension Mechanism
XML JAXP
lang and util Base Libraries
lang and util
Collections
Concurrency Utilities
JAR
Logging
Management
Preferences API
Ref Objects
Reflection
Regular Expressions
Versioning
Zip
Instrument
Java Virtual Machine
Java Hotspot™ Client VM
Java Hotspot™ Server VM
Platforms
Solaris™
Linux
Windows
Other
Java SE API

Install the JDK

- Double-click on the downloaded file from java.sun.com.
- Windows: Modify the PATH environment variable so that javac can be found. (See InstallingJDK movie in ctools or ask us for help).





Control Panel -> System -> Advanced ->
Environment Variables. Edit Path.
Add semi-colon followed by
C:\Program Files\Java\jdk1.6.X_XX\bin

Note: The version may be different from the above

Java Overview: Virtual Machine

- An abstract computing machine that has an instruction set and manipulates memory at run time
- The Java virtual machine is ported to different platforms to provide hardware- and operating system-independence

Hello World Explained

```
// HelloWorld.java

/**

 * Your first Java program.

 */

public class HelloWorld {

  public static void main(String[]
   args) {

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

  }
```

Use any text editor
to create this file
and save it in a 282 folder

Use javac to compile it:
% javac HelloWorld.java

Use java to run it:
% java HelloWorld

Hello World Explained

```
// HelloWorld.java
/**
 * Your first Java program.
 */
public class HelloWorld {
  public static void main(String[] args) {
    System.out.println("Hello World");
  }
}
```

- A Java file name must be the same as the class name
  - Class = `HelloWorld`
  - File name = `HelloWorld.java`
  - Bytecode class file name = `HelloWorld.class`

Hello World Explained

```
// HelloWorld.java
```

```
/**
 * Your first Java program.
 */
public class HelloWorld {
  public static void main(String[] args) {
    System.out.println("Hello World");
  }
}
```

- Java comments come in two forms
  - Line comments: `//`
  - Block comments `/* */`
- Use line comments for single-line comments
- Use block comments for multi-line comments
- There are also Javadoc comments

Hello World Explained

```
// HelloWorld.java
/**
 * Your first Java program.
 */
public class HelloWorld {
  public static void main(String[] args) {
    System.out.println("Hello World");
  }
}
```

- All Java programs exist within classes

Hello World Explained

```
// HelloWorld.java
/**
 * Your first Java program.
 */
public class HelloWorld {
  public static void main(String[] args) {
    System.out.println("Hello World");
  }
}
```

- Almost all Java programs you write start with a call to the main method

Hello World Explained

```
// HelloWorld.java
package edu.um.eecs285.packageexample;

/**
 * Your first Java program.
 */
public class HelloWorld {
  public static void main(String[] args) {
    System.out.println("Hello World");
  }
}
```

- Almost all Java programs you write start with a call to the main method
  - static means the method is part of its class and not part of objects

Hello World Explained

```
// HelloWorld.java
package edu.um.eecs285.packageexample;

/**
 * Your first Java program.
 */
public class HelloWorld {
  public static void main(String[] args) {
    System.out.println("Hello World");
  }
}
```

- Almost all Java programs you write start with a call to the main method
  - static means the method is part of its class and not part of objects
  - void: main does not return a value

Hello World Explained

```
// HelloWorld.java
package edu.um.eecs285.packageexample;

/**
 * Your first Java program.
 */
public class HelloWorld {
  public static void main(String[] args) {
    System.out.println("Hello World");
  }
}
```

- Almost all Java programs you write start with a call to the main method
  - static means the method is part of its class and not part of objects
  - Does not return a value
  - Its parameter is an array of Strings
    - args[0] = the first argument, not the name of the program

Hello World Explained

```
// HelloWorld.java
package edu.um.eecs285.packageexample;


/**
 * Your first Java program.
 */
public class HelloWorld {
  public static void main(String[] args) {
    System.out.println("Hello World");
  }
}
```

- The “Hello, World” is called a string literal
- This is a string that begins and ends with a double quote
- This line is printed to the console

Hello World Explained

```
// HelloWorld.java
package edu.um.eecs285.packageexample;

/**
 * Your first Java program.
 */
public class HelloWorld {
  public static void main(String[] args) {
    System.out.println("Hello World");
  }
}
```

- You use **braces** to enclose, or group multiple programming statements into *compound statements* or *blocks*
- There are many different bracing styles

```
foo() {
}
foo()
{
}
foo()
  {
  }
```

Hello World Explained

```
// HelloWorld.java


/**
 * Your first Java program.
 */
public class HelloWorld {
  public static void main(String[] args) {
    System.out.println("Hello World");
  }
}
```

- Statements are terminated by a **semicolon**

Install Eclipse

- Eclipse needs to be simply unzipped.