



# D06

# PROGRAMMING with JAVA

## Ch1 – Introduction

PowerPoint presentation, created by Angel A. Juan - [ajuanp@gmail.com](mailto:ajuanp@gmail.com),  
for accompanying the book "Big Java", by Cay S. Horstmann (2006)

# Chapter Goals

---

- To review the activity of programming
- To review about **machine code** and high level programming languages
- To become familiar with your computing environment and your compiler
- To compile and run your first **Java** program
- To recognize **syntax** and **logic errors**

# What Is Programming?

---

- Computers are programmed to perform tasks
- Different tasks = different programs
- **Program**
  - Sequence of basic operations executed in succession
  - Contains instruction sequences for all tasks it can execute
- Sophisticated programs require teams of highly skilled programmers and other professionals

# Self Check

---

1. Where is a program stored when it is not currently running?
2. Which part of the computer carries out arithmetic operations, such as addition and multiplication?

# Answers

---

1. In secondary storage, typically a hard disk.
2. The central processing unit.

# Machine Code

- A typical sequence of machine instructions is:
  - Load the contents of memory location 40.
  - Load the value 100.
  - If the first value is greater than the second value, continue with the instruction that is stored in memory location 240.
- Machine instructions are encoded as numbers:

21	40
16	100
163	240

- A **compiler** translates the high-level description into machine instructions for a particular processor
- The **Java Virtual Machine (JVM)** – enables Java applications to run on multiples CPUs and OS - **Bytecode**

# The Java Programming Language

---

- Simpler than C++, Java has become one of the most important languages for general-purpose programming
- Designed for the Internet (applications and applets)
- **Safe** (it has an assortment of security features that guarantees that no evil applets can run on your computer)
- Portable or **platform-independent** ("write once, run anywhere: Unix, Windows, ...")
- Rich **library packages**: graphics, cryptography, networking, database storage, simulation, etc.
- We will use **version 5.0** of Java (or Java 1.5)

# Self Check

---

3. What are the two most important benefits of the Java language?
4. How long does it take to learn the entire Java library?

# Answers

---

3. Safety and portability.
4. No one person can learn the entire library—it is too large.

# Becoming Familiar with your Computer

---

- Download and install:
  - J2SE Development Kit (JDK) 5.0 – [www.sun.com](http://www.sun.com) (includes the Java Runtime Environment)
  - J2SE Development Kit Documentation 5.0
  - A Java **Integrated Development Environment** (JCreator, NetBeans, Eclipse, BlueJ, ...)
- Get familiar with your Java IDE
- Develop a strategy for keeping backup copies of your work

# Editing and compiling in Java

---

- Java has **free-form layout** (you can use any number of spaces and line breaks to separate words)
- Most Java compilers require that Java source files end in an extension “**.java**”
- Java file names cannot contain spaces, and the distinction between upper- and lowercase letters is important (Java is **case sensitive**)

# File HelloTester.java

```
1: public class HelloTester
2: {
3:     public static void main(String[] args)
4:     {
5:         // Display a greeting in the console window
6:
7:         System.out.println("Hello, World!");
8:     }
9: }
```

Starts a new **class**. In Java, every program consists of one or more classes. The keyword **public** denotes that the class is usable by the "public". **Every source file can contain at most one public class, and the name of the public class must match the name of the file containing the class.**

Defines a method called **main()**. A method contains a collection of programming instructions that describe how to carry out a particular task. **Every Java application must have a main() method.** Most Java programs contain other methods besides `main`. The parameter `string[] args` is a required part of the `main` method. The keyword **static** indicates that the `main` method does not operate on an object.

Whenever you **call a method** in Java, you need to specify three items:

1. The **object** that you want to use (`System.out`)
2. The name of the **method** you want to use (`println`)
3. A pair of parentheses, containing any other information (**parameters**) the method needs.

Any text enclosed between `//` and the end of the line is completely ignored by the compiler. The statements in the body of the `main` method (enclosed by `{}`) are executed one by one. Each statement ends in a semicolon `;`.

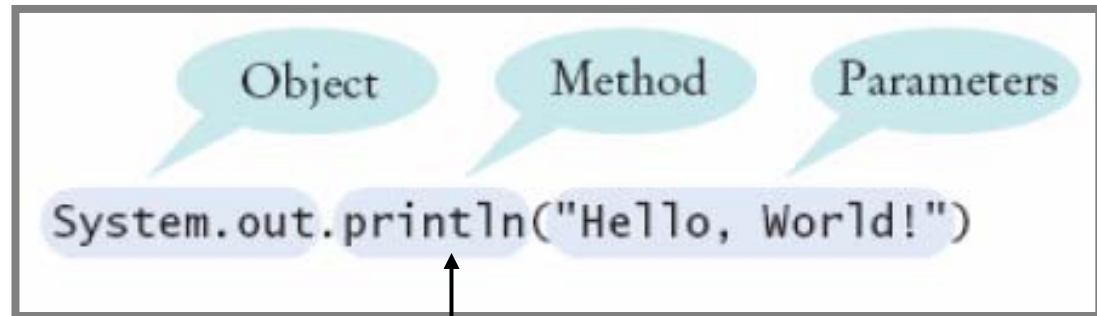


# A Simple Program

- `public class ClassName`
- `public static void main(String[] args)`
- `// comment`
- **Method call**

If you have a comment that is longer than a line, use the `/*...*/` symbols to enclose it.

**Figure 13:  
Calling a Method**



System Class  
System.out Object  
Println() Method

A sequence of characters enclosed in quotation marks is called a **string**. The `println()` method prints a string or a number and then starts a new line. There is another method, called `print()`, that you can use to print an item without starting a new line.

# Syntax 1.1: Method Call

---

*object.methodName(parameters)*

**Example:**

```
System.out.println("Hello, Dave!");
```

**Purpose:**

To invoke a method of an object and supply any additional parameters

# Self Check

---

5. How would you modify the HelloTester program to print the words "Hello, " and "World!" on two lines?
6. Would the program continue to work if you omitted the line starting with //?
7. What does the following set of statements print?

```
System.out.print("My lucky number is");  
System.out.println(3 + 4 + 5);
```

# Answers

---

5. 

```
System.out.println("Hello,");  
System.out.println("World");
```
6. Yes—the line starting with `//` is a comment, intended for human readers. The compiler ignores comments.
7. The printout is `My lucky number is12.`  
It would be a good idea to add a space after the `is`.

# Errors

- Syntax errors

```
System.ouch.print( ". . ." );  
System.out.print( "Hello) ;
```

All Java **keywords** use only lowercase letters. Names of classes usually start with uppercase letters, names of methods and variables with a lowercase letter.

Detected by the compiler

- Logic errors

```
System.out.print( "Hell" );
```

Detected (hopefully) through testing

**Debuggers** are special software tools that let you trace through a program to find bugs –that is, logical errors.

# Self Check

---

8. Suppose you omit the `//` characters from the `HelloTester.java` program but not the remainder of the comment. Will you get a compile-time error or a run-time error?
9. How can you find logic errors in a program?

# Answers

---

8. A compile-time error. The compiler will not know what to do with the word display.
9. You need to run the program and observe its behavior.

# The Compilation Process in Java

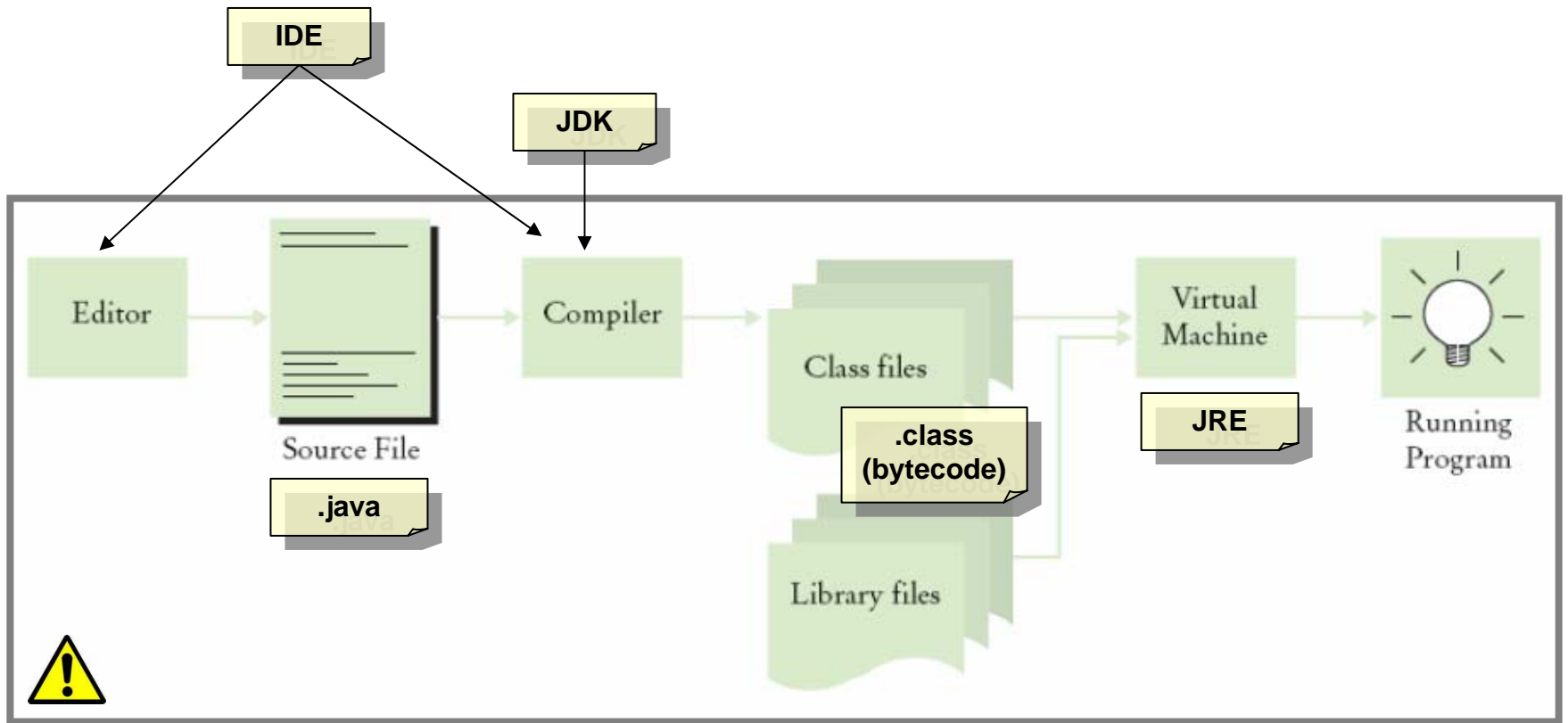


Figure 14:  
From Source Code to Running Program

# Self Check

---

10. What do you expect to see when you load a class file into your text editor?
11. Why can't you test a program for run-time errors when it has compiler errors?

# Answers

---

10. A sequence of random characters, some funny-looking. Class files contain virtual machine instructions (bytecode) that are encoded as binary numbers.
11. When a program has compiler errors, no class file is produced, and there is nothing to run.

# Chapter Summary

---

- The instruction set of the **Java virtual machine** (JVM) can be executed on many CPUs
- Java was originally designed for programming consumer devices
- Java was designed to be **safe** and **portable**. It has a very large library
- Java is **case sensitive**
- Lay out your programs so that they are easy to read
- **Classes** are the fundamental building blocks of Java programs

*Continued...*

# Chapter Summary

---



Every Java application contains a class with a **main()** method. When the application starts, the instructions in the main method are executed

- Each class contains definitions of methods. A **method** is called by specifying an object, the method name, and the method parameters
- A **syntax error** is a violation of the rules of the programming language. A **logic error** causes a program to take an action that the programmer did not intend
- The **Java compiler** translates source code into class files that contain instructions for the Java virtual machine. The JVM loads program instructions from class files and library files