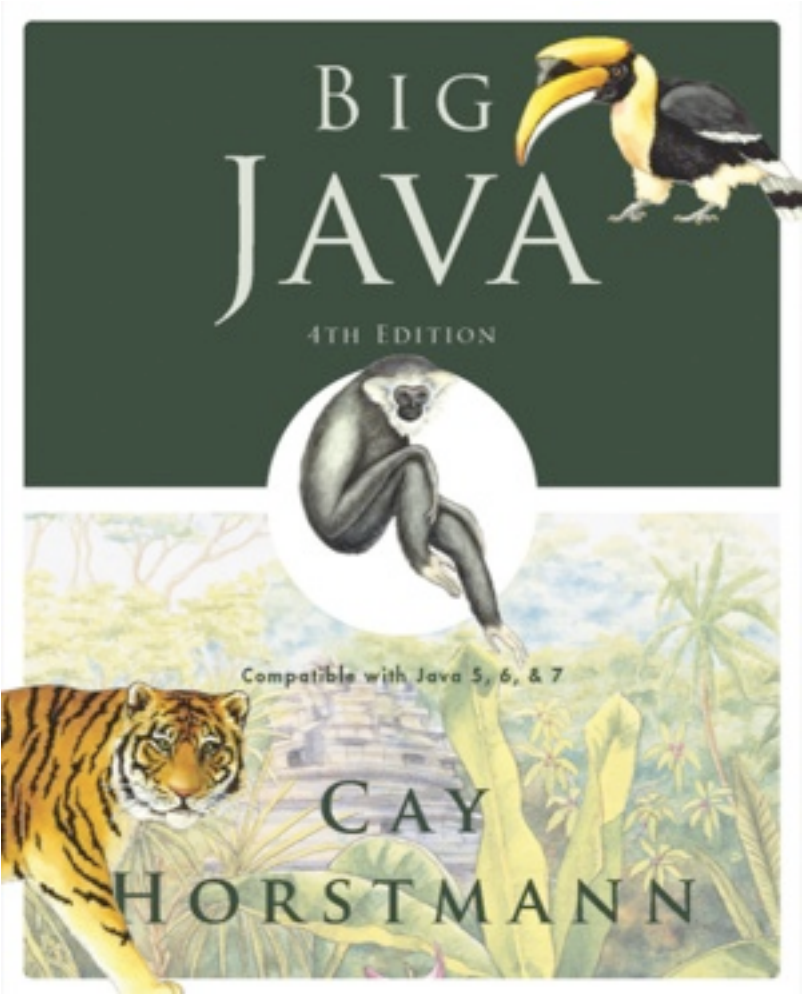


Chapter 1 – Introduction

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

The Java Programming Language

- Simple
- Safe
- Platform-independent (“write once, run anywhere”)
- Rich library (packages)
- Designed for the internet

ch01/hello/HelloPrinter.java

```
1  public class HelloPrinter
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  }
```

Program Run:

```
Hello, World!
```

The Structure of a Simple Program: Class Declaration

- Classes are the fundamental building blocks of Java programs:

  ```
  public class HelloPrinter
  ```

  starts a new **class**

- Every source file can contain at most one public class
- The name of the public class must match the name of the file containing the class:
  - *Class* `HelloPrinter` *must be contained in a file named* `HelloPrinter.java`

The Structure of a Simple Program: `main` Method

- Every Java application contains a class with a main method
  - *When the application starts, the instructions in the main method are executed*
-
```
public static void main(String[] args)
{
   . . .
}
```

  declares a `main` method

The Structure of a Simple Program: Comments

- The first line inside the main method is a comment:

```
// Display a greeting in the console window
```

- Compiler ignores any text enclosed between `//` and end of the line
- Use comments to help human readers understand your program

The Structure of a Simple Program: Statements

- The body of the main method contains statements inside the curly brackets (`{}`)
- Each statement ends in a semicolon (`;`)
- Statements are executed one by one
- Our method has a single statement:

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

which prints a line of text:

```
Hello, World
```

The Structure of a Simple Program: Method Call

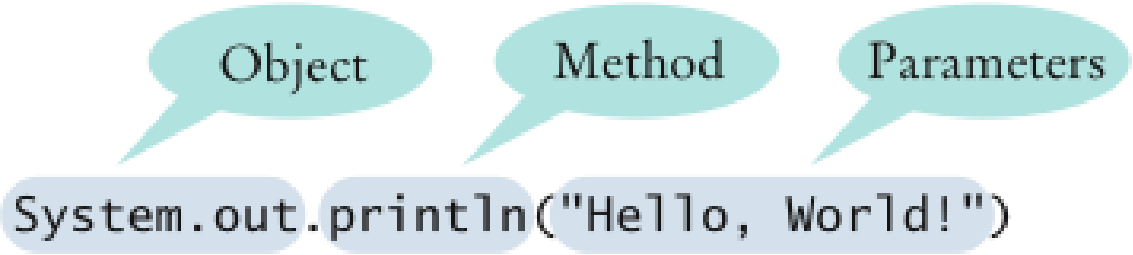
- `System.out.println("Hello, World!");`

  is a *method call*

- A method call requires:

  1. *The object that you want to use (in this case, `System.out`)*
  2. *The name of the method you want to use (in this case, `println`)*
  3. ***Parameters** enclosed in parentheses (`( )`) containing any other information the method needs (in this case, `"Hello, World!"`)*

Figure 7
Calling a Method



Syntax 1.1 Method Call

Syntax *object.methodName(parameters)*

Example

The method is invoked on this object.

This is the name of the method.

This parameter is passed to the method.

```
System.out.println("Hello")
```

Parameters are enclosed in parentheses.
Multiple parameters are separated by commas.

The Structure of a Simple Program: Strings

- **String:** a sequence of characters enclosed in double quotation marks:

```
"Hello, World!"
```

Editing a Java Program

- Use an editor to enter and modify the program text
- Java is case-sensitive
  - *Be careful to distinguish between upper- and lowercase letters*
- Lay out your programs so that they are easy to read

Compiling and Running a Java Program

- The Java compiler translates source code into class files that contain instructions for the Java virtual machine
- A class file has extension `.class`
- The compiler does not produce a class file if it has found errors in your program
- The Java virtual machine loads instructions from the program's class file, starts the program, and loads the necessary library files as they are required

`HelloPrinter` in a Console Window

```
Terminal
File Edit View Terminal Tabs Help
~$ cd BigJava/ch01/hello
~/BigJava/ch01/hello$ javac HelloPrinter.java
~/BigJava/ch01/hello$ java HelloPrinter
Hello, World!
~/BigJava/ch01/hello$
```

Figure 8 Running the `HelloPrinter` Program in a Console Window

`HelloPrinter` in an IDE

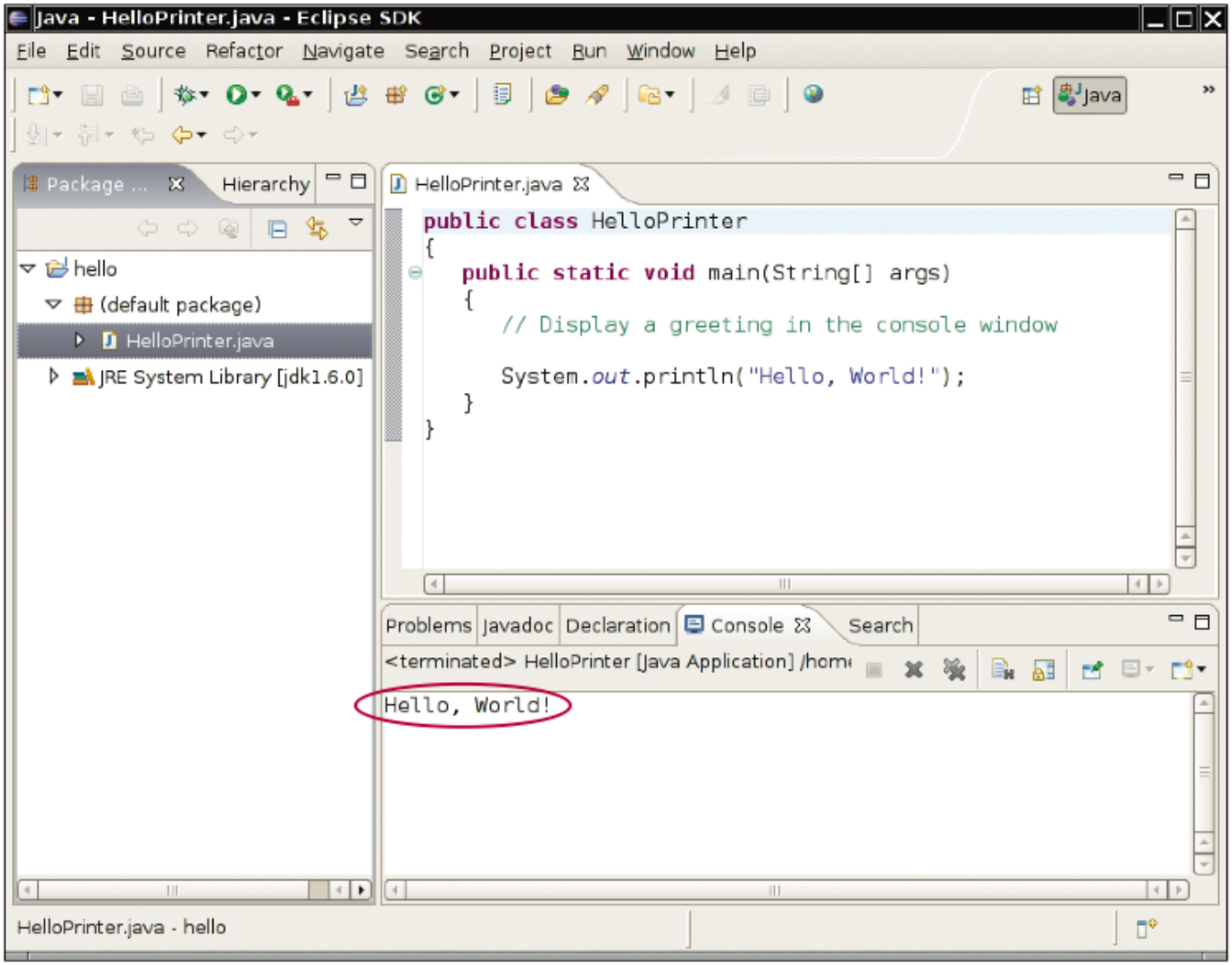


Figure 9 Running the `HelloPrinter` Program in an Integrated Development Environment

From Source Code to Running Program

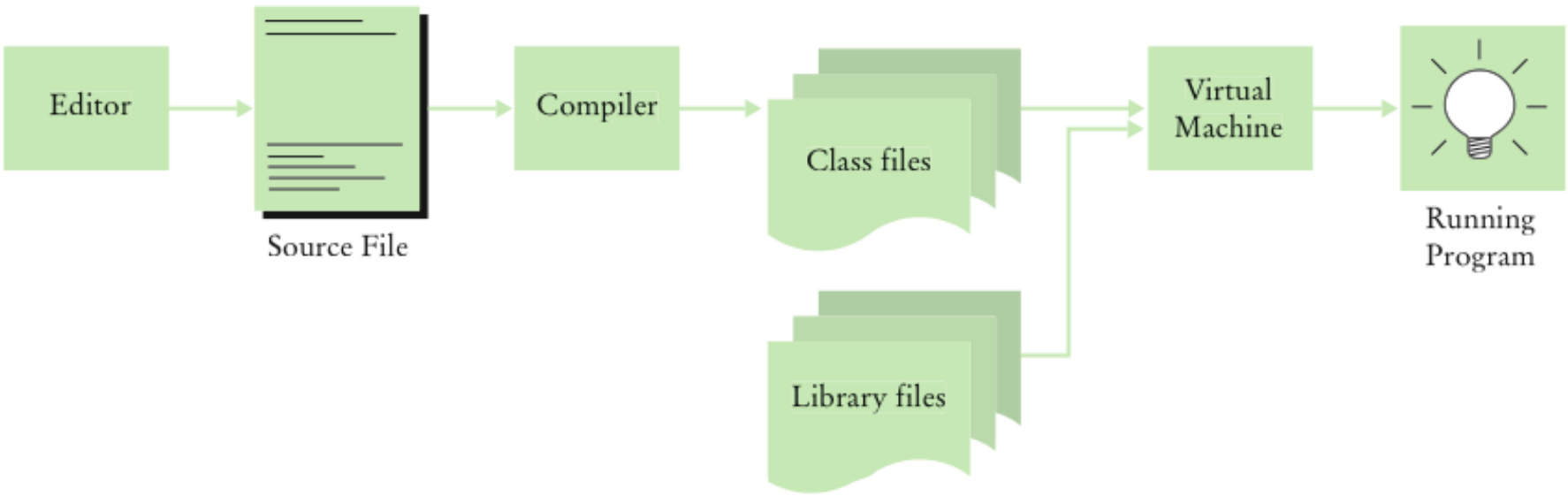


Figure 10 From Source Code to Running Program

Errors

- **Compile-time error:** A violation of the programming language rules that is detected by the compiler
  - *Example*:

```
System.ou.println("Hello, World!);
```

  - *Syntax error*
- **Run-time error:** Causes the program to take an action that the programmer did not intend
  - *Examples:*

```
System.out.println("Hello, Word!");
System.out.println(1/0);
```

  - *Logic error*