Dasar-Dasar Pemrograman 2: Java Basics

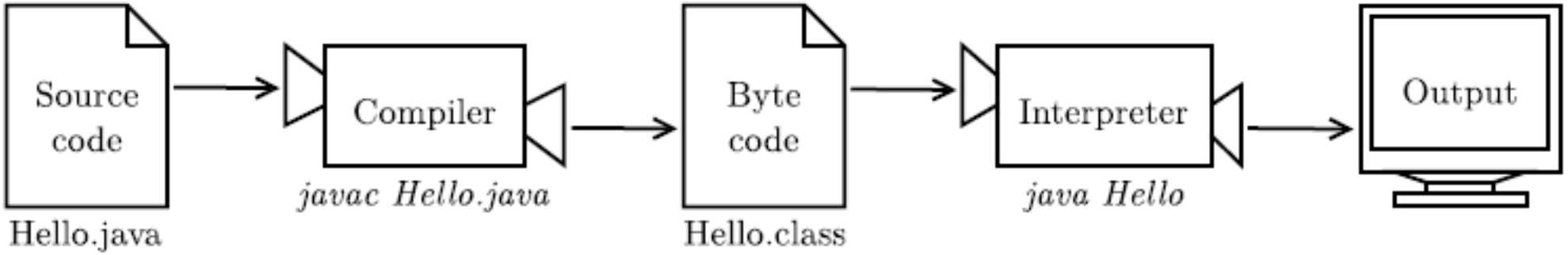
Fariz Darari (fariz@cs.ui.ac.id)

UNIVERSITAS INDONESIA
FAKULTAS ILMU KOMPUTER

Problem solving

- The single most important skill for a computer scientist is **problem solving**.
- It's the ability to:
  - formulate problems,
  - think creatively about solutions, and
  - express solutions clearly and accurately.
- As it turns out, the process of learning to program is an excellent opportunity to develop problem solving skills.

Compiling and running a Java program



- Java is both compiled and interpreted.
- Instead of translating programs directly into machine language, the Java compiler generates byte code.
- Similar to machine language, byte code is easy and fast to interpret.
- But it is also portable, so it is possible to compile a Java program on one machine, transfer the byte code to another machine, and run the byte code on the other machine.
- The interpreter that runs byte code is called a "Java Virtual Machine" (JVM).

Hello, World!

```
public class Hello {

    public static void main(String[] args) {
        // generate some simple output
        System.out.println("Hello, World!");
    }
}
```

When this program runs it displays:

```
Hello, World!
```

Escape sequences

```
public class Hello {

    public static void main(String[] args) {
        System.out.print("Hello!\nHow are you doing?\n");
    }
}
```

The output is two lines, each ending with a newline character:

```
Hello!
How are you doing?
```

Escape sequences

`\n`	newline
`\t`	tab
`\"`	double quote
`\\`	backslash

Table 1.1: Common escape sequences

We could write something like this, but..

```
public class Goodbye { public static void main(String[] args)
{ System.out.print("Goodbye, "); System.out.println
("cruel world");}}
```

We could write something like this, but..

```
public class Goodbye { public static void main(String[] args)
{ System.out.print("Goodbye, "); System.out.println
("cruel world");}}
```

This one reads better :)

```
public class Goodbye {

    public static void main(String[] args) {
        System.out.print("Goodbye, ");
        System.out.println("cruel world");
    }
}
```

Quiz time

- What is a program?
- What is a bug?
- What is compile?

Quiz time

- What is a program?

A sequence of instructions that specifies how to perform tasks on a computer.

- What is a bug?

An error in a program.

- What is compile?

To translate a program in a high-level language into a low-level language, **all at once**, in preparation for later execution.

Variables

- A variable is a named location that stores a value. Values may be numbers, text, images, sounds, and other types of data.

```
String message;
```

This statement is a **declaration**, because it declares that the variable named `message` has the type `String`. Each variable has a **type** that determines what kind of values it can store. For example, the `int` type can store integers, and the `char` type can store characters.

Declaration

```
int hour, minute;
```

Assignment

```
hour = 11;              // assign the value 11 to hour
minute = 59;            // set minute to 59
```

Declare and assign variables on same line

```
String message = "Hello!";
int hour = 11;
int minute = 59;
```

Printing variables

```
int hour = 11;
int minute = 59;
System.out.print("The current time is ");
System.out.print(hour);
System.out.print(":");
System.out.print(minute);
System.out.println(".");
```

The output of this program is:

```
The current time is 11:59.
```

Arithmetic operators

The following program converts a time of day to minutes:

```
int hour = 11;
int minute = 59;
System.out.print("Number of minutes since midnight: ");
System.out.println(hour * 60 + minute);
```

In this program, **`hour * 60 + minute`** is an **expression**, which represents a single value to be computed. When the program runs, each variable is replaced by its current value, and then the operators are applied. The values operators work with are called **operands**.

The result of the previous example is:

```
Number of minutes since midnight: 719
```

Floating-point numbers

```
double pi;
pi = 3.14159;
```

```
double minute = 59.0;
System.out.print("Fraction of the hour that has passed: ");
System.out.println(minute / 60.0);
```

The output is:

```
Fraction of the hour that has passed: 0.9833333333333333
```

Error types

- **Compile-time error**

Compile-time errors occur when you violate the syntax rules of the Java language. For example, parentheses and braces have to come in matching pairs. So **(1 + 2)** is legal, but **8)** is not.

- **Runtime error**

The second type of error is a run-time error, so-called because it does not appear until after the program has started running.

- **Logic error**

If your program has a logic error, it will compile and run without generating error messages, but it will not do the right thing.

Credits: Chapter 1 and 2 of Think Java book by Allen Downey and Chris Mayfield