Java Actually: Section 2.7 Review Questions

COMP SCI / SFWR ENG 2S03

Natalie Perna

Department of Computing and Software
McMaster University

Week 3: Sept 24 - 28

Outline

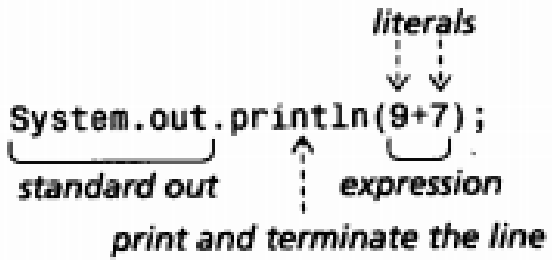
Resource

All of the following material is adapted from:
Mughal, Khalid. Java actually : a comprehensive primer in programming. Australia: Course Technology/Cenage Learning, 2008.

Literals and Expressions

Literal: Value written directly in the source code (e.g. 9)
Expression: Always evaluates to a value (e.g. 9 + 7)
String (Literal): Sequence of characters enclosed in double quotes (e.g. "The expression 9 + 7 evaluates to 16.")

FIGURE 2.1 Printing the sum of two integers to the terminal window



Addition and Concatenation

Plus (+) operator is used for **addition** and **concatenation**.

Concatenation: Contents of two strings appended (joined) resulting in a new string

```
System.out.println(1 + 1);
System.out.println("Hello, my name is " + "Natalie.");
```

↓

```
2
Hello, my name is Natalie.
```

2.7.1 Review Question

What do you think the following method calls would print to the terminal window?

```
System.out.println(10 + 10 + 20);
System.out.println("10+10 is " + 20);
System.out.println("10+10 is " + (10 + 10));
System.out.println("10+10 is " + 10 + 10);
```

2.7.1 Review Question

What do the method calls actually show?

```
System.out.println(10 + 10 + 20);
```

↓

```
40
```

2.7.1 Review Question

```
System.out.println("10+10 is "+20);
```

↓

```
10+10 is 20
```

2.7.1 Review Question

```
System.out.println("10+10 is "+(10+10));
```

↓

```
10+10 is 20
```

2.7.1 Review Question

```
System.out.println("10+10␣is␣"␣+␣10␣+␣10);
```

↓

Left to Right: The string "10+10 is " is first concatenated with the string representation of the value 10, then the resulting string "10+10 is 10" is concatenated with the string representation of the value 10.

↓

```
10+10␣is␣1010
```

Choosing Names

Remember two things:

1. rules for what Java accepts as valid names
2. conventions for variable names

Rules

Variable names:

- Can contain letters, digits, and underscores
- Cannot start with a digit
- Are case sensitive
- Cannot be keywords (e.g. int, if, while) (See Table B.1 on page 725 for complete list)

Conventions

Variable names:

- Should not contain more than 15-20 characters
- Start with a lower-case letter and use all lower case letters except for the first letter of each consecutive word
- Except if they are constants, then use all upper-case letters with underscores to separate words
- See Tutorial 2 slides for more.

2.7.5 Review Question

Which of the following variable names are valid in Java? Which adhere to conventions? Which variables have meaningful names? Justify your answers.

```
a) minimum-Price
b) minimumPrice
c) XYZ
d) xCoordinate
e) y2k
f) isDone
g) numberOfDaysInALeapYear
h) JDK_1_6_0
```

2.7.5 Review Question

a) minimum-Price
Invalid. Variable names cannot contain dashes.

b) minimumPrice
Valid. Adheres to conventions. Meaningful.

c) XYZ
Valid. Adheres to conventions for a constant. Probably not meaningful.

d) xCoordinate
Valid. Adheres to conventions. Meaningful.

2.7.5 Review Question

e) y2k
Valid. Does not use complete words. Might be meaningful (depends on context).

f) isDone
Valid. Adheres to conventions. Meaningful.

g) numberOfDaysInALeapYear
Valid. Too long.

h) JDK_1_6_0
Valid. Adheres to conventions for a constant. Probably not meaningful.

printf()

```
printf(String format, Object... args)
```

format:

- Contains format specifications
- Determines how each subsequent value in the parameter args will be formatted and printed

Object... args: Method accepts zero or more parameters

Format Specifications I

Table 2.2 Format specifications in Java (Page 32)

Parameter value	Format specification	Example value	String printed	Comment
Integer	"%d"	125	"125"	Occupies as many character places as needed.
	"%6d"	125	" 125"	Occupies six character places and is right-justified. The printed string is padded with spaces to the left.
	"%02d"	3	"03"	Occupies two character places and is padded with leading zeros.

Format Specifications II

Parameter value	Format specification	Example value	String printed	Comment
Floating point value	`"%f"`	`16.746`	`"16.746000"`	Occupies as many character places as needed, but always includes six decimal places.
	`"%.2f"`	`16.746`	`"16.75"`	Occupies as many character places as needed, but includes only two decimal places.
	`"%8.2f"`	`16.7466`	`"  16.75"`	Occupies eight character places, including the decimal point, and uses two decimal places.

Format Specifications III

Parameter value	Format specification	Example value	String printed	Comment
String	`"%s"`	`"Hi!"`	`"Hi!"`	Occupies as many character places as are needed.
	`"%12s"`	`"Hi Dude!"`	`"  Hi Dude!"`	Occupies twelve character places and is right-justified.
	`"%-12s"`	`"Hi Dude!"`	`"Hi Dude!  "`	Occupies twelve character places and is left-justified.
Linefeed	`"%n"`	(none)	(none)	Moves the cursor to the next line in the terminal window.

printf(): Example

The following calls to the printf() method:

```
System.out.printf("Player\\Game␣␣␣␣␣␣␣␣␣␣%6d%6d%6d%n",
1,␣2,␣3);
System.out.printf("%-20s%6d%6d%6d%n",␣"F.␣Reshmann",
320,␣160,␣235);
System.out.printf("%-20s%6d%6d%6d%n",␣"A.␣King",
1250,␣1875,␣2500);
```

will generate this tabular printout of game results:

```
Player\Game              1     2     3
F. Reshmann            320   160   235
A. King               1250  1875  2500
```

2.7.11 Review Question

Use the System.out.printf() method to print the following values:

a) A six-digit integer, including the sign, e.g. 123456 as +123456.

b) The floating-point value 123456789.3837 in scientific notation, i.e. as 1.234568e+08.

c) The string "We are 100% motivated to learn Java!".

d) The number 1024 as a right-justified eight-digit integer, i.e. as 00001024.

2.7.11 Review Question

a) A six-digit integer, including the sign, e.g. 123456 as +123456.

```
System.out.printf("%+6d", 123456);
```

↓

```
+123456
```

2.7.11 Review Question

b) The floating-point value 123456789.3837 in scientific notation, i.e. as 1.234568e+08.

```
System.out.printf("%e", 123456789.3837);
```

↓

```
1.234568e+08
```

2.7.11 Review Question

c) The string "We are 100% motivated to learn Java!".

```
System.out.printf("%s%d%%%s", "We are ", 100, " motivated to learn Java!");
```

↓

```
We are 100% motivated to learn Java!
```

2.7.11 Review Question

d) The number 1024 as a right-justified eight-digit integer, i.e. as 00001024.

```
System.out.printf("%08d", 1024);
```

↓

```
00001024
```