Formatted Output / User IO

COMP SCI / SFWR ENG 2S03

Dylan Aspden

Department of Computing and Software
McMaster University

Week 4: Sept 30 - Oct 4

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.

Special thanks to Natalie Perna for slides on formatted printing

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.

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.

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!".

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.

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
```

Scanner Class

- Class from the Java standard library which provides the ability to read numbers and strings from the keyboard
- Must first tell the Java compiler that we wish to use the class
  - Done through an import statement

```
import java.util.Scanner;
```

Scanner Class: Methods

- Scanner class provides a number of methods for retrieving input from the keyboard

```
Scanner keyboard = new Scanner(System.in);
int i = keyboard.nextInt();
double d = keyboard.nextDouble();
String s = keyboard.next();
```

- With these methods, we can take input from a keyboard, store it inside our program, and finally use it in our computations

Reading from Keyboard: Example #1

```
import java.util.Scanner;
public class IntegerReader {
  public static void main(String[] args) {
    Scanner keyboard = new Scanner(System.in);
    System.out.print("Enter an integer: ");
    int numberRead = keyboard.nextInt();
    System.out.printf("You entered: %d%n", numberRead);
  }
}
```

```
Program output:
Enter an integer: 123
You entered: 123
```

Reading from Keyboard: Example #2

```
import java.util.Scanner;
public class FloatingPointArea {
  public static void main(String[] args) {
    Scanner keyboard = new Scanner(System.in);

    System.out.print("Enter␣dimensions␣(l,␣w):␣");
    double length = keyboard.nextDouble();
    System.out.print("␣");
    double breadth = keyboard.nextDouble();
    double area = length * breadth;

    System.out.printf(
      "A␣rectangle␣of␣length␣%.2f␣cm␣and␣breadth␣%.2f␣cm" +
      "␣has␣area␣%.2f␣sq.␣cm.␣%n",
      length, breadth, area);
  }
}
```

Reading from Keyboard: Example #2

```
Program output:
Enter dimensions (l, w): 15.5 4.25
A rectangle of length 15.50 cm and breadth 4.25 cm has area
  65.88 sq. cm.
```

→ Note that the line wrap was due to space constraints. In reality this output would only span one line in the console.

Remarks

Best Practice

Provide meaningful prompts to make the use of the program self-explanatory. The prompt should be short, enabling a user to enter the required value on the same line. *(Mughal, 2008)*

Exercises

Please try these exercises from the book in order to familiarize yourselves with keyboard I/O in Java:

- (2.8.5)
- (2.8.9)

Exercise #1

Write a program that calculates the area a and circumference c of a circle:

$$a = p \times r^2$$

$$c = 2 \times p \times r$$

where p is a mathematical constant, and r is the radius of the circle. The square of the radius can be calculated by multiplying r with itself. Assume a value of 3.1415927 for p (π), which should be defined as a constant in your program.

Exercise #2

Extend the program in Exercise #1 to read the units measurement for the radius, e.g. *cm*, *m*, or *km*, as a string. You can use the *nextLine() / next()* methods for this purpose.

Now run the program to calculate the circumference of the Earth at equator, when the equatorial radius is 6378.135 *km*.

End

Questions / Comments / Concerns?