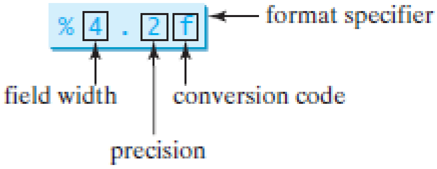
AP Computer Science
Chapter 9 Notes (6)

Name: ______________________________

FORMATTING CONSOLE OUTPUT

You can use the `System.out.printf()` method to display formatted output on the console.

```
double amount = 12618.98;
double interestRate = 0.0013;
double interest = amount * interestRate;
System.out.printf("Interest is %4.2f", interest);
```



The syntax to invoke this method is

```
System.out.printf(format, item1, item2, ..., itemk)
```

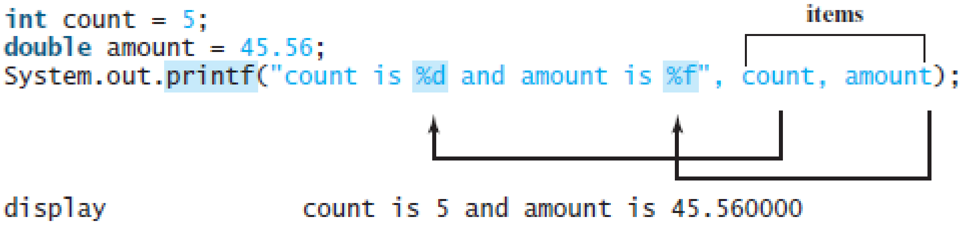
where format is a string that may consist of substrings and format specifiers.

A format specifier specifies how an item should be displayed. An item may be a numeric value, character, a Boolean value, or a string. A simple format specifier consists of a percent sign (`%`) followed by a conversion code.

Frequently Used Format Specifiers

Format Specifier	*Output*	*Example*
%b	a Boolean value	true or false
%c	a character	‘a’
%d	a decimal integer	200
%f	a floating-point number	45.460000
%e	a number in standard scientific notation	4.556000e+01
%s	a string	“Java is cool”

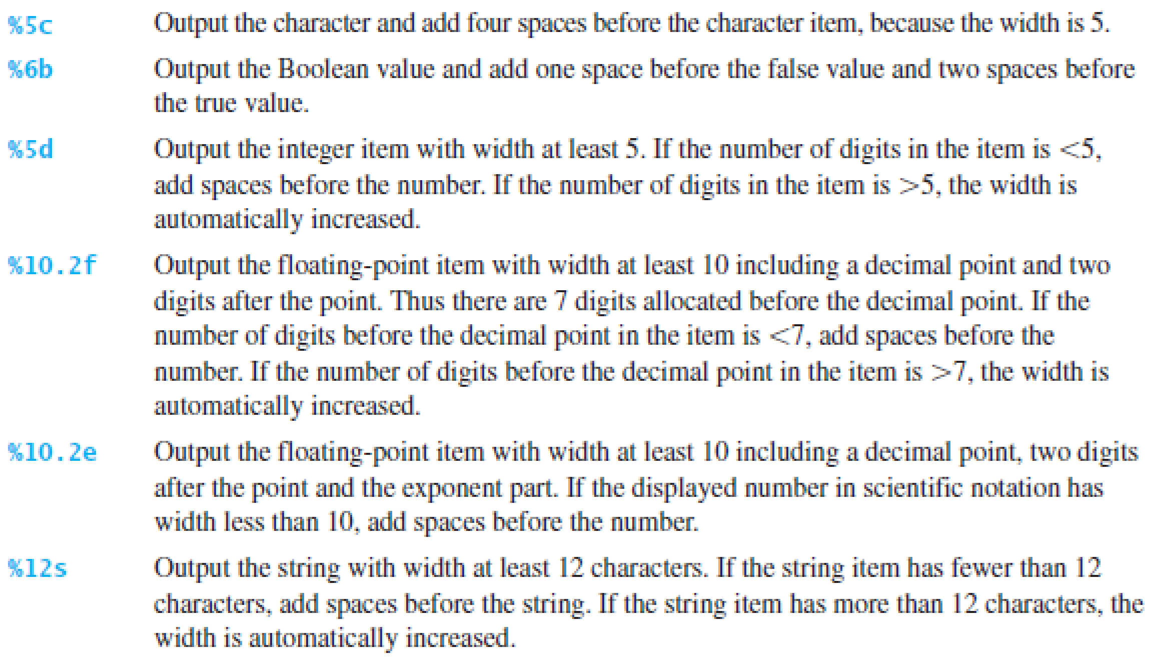
Example:



```
int count = 5;
double amount = 45.56;
System.out.printf("count is %d and amount is %f", count, amount);

display              count is 5 and amount is 45.560000
```

Items must match the format specifiers in order, in number, and in exact type. For example, the format specifier for count is `%d` and for amount is `%f`. By default, a floating-point value is displayed with six digits after the decimal point. You can specify the width and precision in a format specific as shown below:



%5c	Output the character and add four spaces before the character item, because the width is 5.
%6b	Output the Boolean value and add one space before the false value and two spaces before the true value.
%5d	Output the integer item with width at least 5. If the number of digits in the item is <5, add spaces before the number. If the number of digits in the item is >5, the width is automatically increased.
%10.2f	Output the floating-point item with width at least 10 including a decimal point and two digits after the point. Thus there are 7 digits allocated before the decimal point. If the number of digits before the decimal point in the item is <7, add spaces before the number. If the number of digits before the decimal point in the item is >7, the width is automatically increased.
%10.2e	Output the floating-point item with width at least 10 including a decimal point, two digits after the point and the exponent part. If the displayed number in scientific notation has width less than 10, add spaces before the number.
%12s	Output the string with width at least 12 characters. If the string item has fewer than 12 characters, add spaces before the string. If the string item has more than 12 characters, the width is automatically increased.

If an item requires more spaces than the specified width, the width is automatically increased. For example, the following code

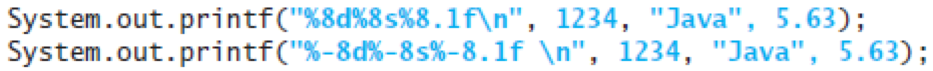
```
System.out.printf("%3d#%2s#%3.2f\n", 1234, "Java", 51.6653);
```

displays

```
1234#Java#51.67
```

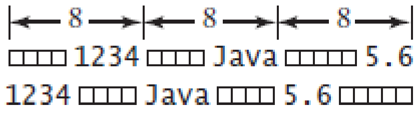
The specified width for `int` item `1234` is 3, which is smaller than its actual size 4. The width is automatically increased to 4. The specified width for string item `Java` is 2, which is smaller than its actual size 4. The width is automatically increased to 4. The specified width for double item `51.6653` is 3, but it needs width 5 to display `51.67`, so the width is automatically increased to 5.

By default, the output is right justified. You can put the minus sign (–) in the format specifier to specify that the item is left justified in the output within the specified field. For example, the following statements



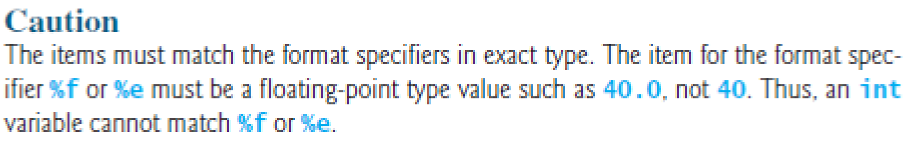
```
System.out.printf("%8d%8s%8.1f\n", 1234, "Java", 5.63);
System.out.printf("%-8d%-8s%-8.1f \n", 1234, "Java", 5.63);
```

display



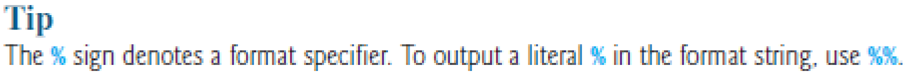
```
□□□□1234□□□□Java□□□□□5.6
1234□□□□Java□□□□5.6□□□□□
```

where the square box denotes a blank space.



Caution

The items must match the format specifiers in exact type. The item for the format specifier `%f` or `%e` must be a floating-point type value such as `40.0`, not `40`. Thus, an `int` variable cannot match `%f` or `%e`.



Tip

The `%` sign denotes a format specifier. To output a literal `%` in the format string, use `%%`.