Java Mickey's Multiple Unit Converter

Contents

Time required: 90 minutes

- Comment each line of code as shown in the tutorials and other code examples.
- Follow all directions carefully and accurately.
- Think of the directions as minimum requirements.

Pseudocode

1. Write pseudocode or TODO for the exercise
2. Submit with the assignment

Formatting Numbers

When you output a double in Java by using print or **println**, it displays up to 16 digits.

```
double division = 4.0 / 3.0;
System.out.println(division);
```

The result is as follows:

```
1.3333333333333333
```

That might be more than you want. **System.out** provides another method, called **printf**, that gives you more control of the format. The "f" in printf stands for "formatted". Here's an example.

```
System.out.printf("Four thirds = %.3f", division);
```

The first value in the parentheses is a format string that specifies how the output should be displayed. This format string contains ordinary text followed by a format specifier, which is a special sequence that starts with a percent sign. The format specifier %.3f indicates that the following value should be displayed as floating-point, rounded to three decimal places:

```
Four thirds = 1.333
```

The format string can contain any number of format specifiers; here's an example with two of them:

```
int inch = 100;
double cm = inch * CM_PER_INCH;
System.out.printf("%d in = %f cm\n", inch, cm);
```

The result is as follows:

```
100 in = 254.000000 cm
```

Like print, printf does not append a newline. \n is needed to add a newline character.

The format specifier %d displays integer values ("d" stands for "decimal", meaning base 10 integer). The values are matched up with the format specifiers in order, so inch is displayed using %d, and cm is displayed using %f.

Learning about format strings is like learning a sublanguage within Java. There are many options, and the details can be overwhelming. The table below lists a few common uses, to give you an idea of how things work.

%d	Integer in base 10 ("decimal")	12345
%,d	Integer with comma separators	12,345
%08d	Padded with zeros, at least 8 digits wide	00012345
%f	Floating-point number	6.789000
%.2f	Rounded to 2 decimal places	6.79
%s	String of characters	"Hello"
%x	Integer in base 16 ("hexadecimal")	bc614e

Assignment: Unit Converter

This program will start by asking the user what type of conversion they wish.

Create a program named **UnitConverter.java**

From a menu, the user can choose from the following conversions. Each conversion will have a separate method.

- Cm to Inches
- Inches to Cm
- Km to Miles
- Miles to Km

Requirements

- Round the results to 2 decimal places.
- You will want to use a while loop menu as we have done in the past.
- If the user enters a negative length, the program should tell the user that the entry is invalid.
- Otherwise, the program should convert the length and print out the result.

Example run:

```
 +---------------------------------------------+
 |           Multiple Unit Converter           |
 +---------------------------------------------+
Choose a conversion
(1) Centimeters --> Inches
(2) Inches --> Centimeters
(3) Kilometers --> Miles
(4) Miles --> Kilometers
Enter your choice: 1
Enter centimeters: 25.3
25.3 centimeters is 9.96 inches.
Another conversion?
(1) Yes (2) to quit) 1
Choose a conversion
(1) Centimeters --> Inches
(2) Inches --> Centimeters
(3) Kilometers --> Miles
(4) Miles --> Kilometers
Enter your choice: 4
Enter miles: 25.6
25.6 miles is 41.20 kilometers.
Another conversion?
(1) Yes (2) to quit) 1
Choose a conversion
(1) Centimeters --> Inches
(2) Inches --> Centimeters
(3) Kilometers --> Miles
(4) Miles --> Kilometers
Enter your choice: 2
Enter inches: 25
25.0 inches is 63.50 centimeters.
Another conversion?
(1) Yes (2) to quit) 1
Choose a conversion
(1) Centimeters --> Inches
(2) Inches --> Centimeters
(3) Kilometers --> Miles
(4) Miles --> Kilometers
Enter your choice: 3
Enter kilometers: 25
25.0 kilometers is 15.53  miles.
Another conversion?
(1) Yes (2) to quit) 2
Bye!
```

Assignment Submission

1. Attach the pseudocode.

2. Attach the program files.
3. Attach screenshots showing the successful operation of the program.
4. Submit in Blackboard.