

# Chapter 3

## Input and Output

### 3.1 Exercises

**Exercise 3.1** Modify the program you wrote in Exercise 2.1 to prompt the user for their age in years.

**Exercise 3.2** Modify the program you wrote in Exercise 2.2 to prompt the user for the air temperature and relative humidity. Use `printf` to display the dew point with one digit to the right of the decimal point.

**Exercise 3.3** Modify the program you wrote in Exercise 2.3 to prompt the user for the air temperature and relative humidity. Use `printf` to display the heat index temperature with one digit to the right of the decimal point.

**Exercise 3.4** In the United States, the “fuel efficiency” of an electric vehicle is measured in miles per kilowatt-hour (kWh). In Canada and Europe, it is measured in kWh per 100km. Write a program that prompts the user for efficiency in miles/kWh . The program will then calculate and display the equivalent efficiency in terms of kWh per/100km. One kilometer equals 0.621371 mile (or, if you prefer, one mile equals 1.609341 km). Here is what the output might look like:

```
Enter miles per kilowatt-hour: 4.5
That is equivalent to 13.81 kWh/100km.
```

**Exercise 3.5** In Canada and Europe, the “fuel efficiency” of an electric vehicle is measured in kWh per 100km. In the United States, it is measured in miles per kilowatt-hour (kWh). Write a program that prompts the user for electric vehicle efficiency in kWh/100km and calculates the equivalent in terms of miles/kWh. One kilometer equals 0.621371 mile (or, if you prefer, one mile equals 1.609341 km). Here is what the output might look like.

```
Enter kilowatt-hours per 100km: 13.5
That is equivalent to 4.60 miles/kWh.
```

**Exercise 3.6** The goal of this exercise is to calculate the price of an order given an item’s price, the quantity purchased, and the sales tax. When it’s finished, it should work like this:

```
Please enter the unit price of the item $2.45
Enter the number you wish to purchase: 5

Subtotal:      $12.25
Tax at 7.50%:   0.92
Total:         $13.17
```

Note: the dollar sign in the first line is part of the prompt; the user does not enter the dollar sign. Use `printf` to display exactly two digits to the right of the decimal point.

**Exercise 3.7** Write a program named `Fluid.java` that calculates the velocity of fluid in a pipe. Ask the user for the flow rate in liters per minute and the pipe diameter in millimeters. The velocity (in meters per second) is calculated with this formula:

$$v = \frac{\text{flow rate} \times 21.22}{\text{diameter}^2}$$

Format the output to two decimal places, and remember to label your input prompt and output. Here is a sample run of the program.

```
Enter flow rate in liters per minute: 7.3
Enter pipe diameter in millimeters: 11.5
Fluid velocity is 1.17 meters per second.
```

**Exercise 3.8** Because the `Scanner` class views input as a stream of characters, you can ask users for multiple inputs with a single prompt. For example, instead of asking the user for the width and height of a rectangle as two inputs in this code fragment:

```
// ask for length in cm
System.out.print("Enter width of rectangle in cm: ");
double width = input.nextDouble();

System.out.print("Enter height of rectangle in cm: ");
double height = input.nextDouble();
```

Which looks like this when the program runs:

```
Enter width of rectangle in cm: 5.3
Enter height of rectangle in cm: 2.6
```

You could use one prompt:

```
System.out.println("Enter width and height of rectangle");
System.out.print("in cm separated by spaces: ");
double width = input.nextDouble();
double height = input.nextDouble();
```

Which looks like this when the program runs:

```
Enter width and height of rectangle
in cm separated by spaces: 5.3 2.6
```

In this exercise, you will prompt the user to enter an angle as degrees, minutes, and seconds and convert it to decimal degrees. Here is what the output might look like:

```
Enter degrees, minutes, and seconds
separated by spaces: 25 15 30
That is 25.2583 degrees (decimal).
```

You may use either one prompt, as shown here, or three separate prompts.

**Exercise 3.9** This is another exercise that uses input and arithmetic operators to calculate the stopping distance for a vehicle. Write a program that prompts the user to enter:

- The initial velocity in kilometers per hour (**double**). A velocity of 60 miles per hour works out to about 97 kilometers per hour.
- The reaction time in seconds (**double**). For most people, the reaction time is about 1.5 seconds.
- The coefficient of friction for the tires (**double**). This is a dimensionless number, normally about 0.7.

Calculate and display the stopping distance using this formula:

$$d = v \cdot t + \frac{v^2}{2\mu g}$$

where  $t$  is the reaction time,  $v$  is the initial velocity in **meters per second**,  $\mu$  is the coefficient of friction between the tires and the roadway, and  $g$  is the acceleration of gravity in meters per second squared. Make  $g$  a constant with the value 9.80665.

Notice that, since you are asking for the velocity in kilometers per hour, you must convert it to meters per second. Define constants `METERS_PER_KM` (1000) and `SECONDS_PER_HOUR` (3600) to use in this calculation.

Here is an example of what the program should look like:

```
Stopping Distance Calculator
Enter velocity of vehicle in kilometers per hour: 97
Enter reaction time in seconds (normal is 1.5): 1.1
Enter coefficient of friction (from 0.1 to 1.0; normal 0.7): 0.5

Stopping distance is 103 meters.
```

In this program, it is better to use three separate prompts for the input because, unlike Exercise 3.8, the entries are not closely related to each other.

**Exercise 3.10** In business, you often want to find your *break-even quantity*: the minimum number of items you must sell before you start making a profit. To calculate the break-even quantity, you must know:

- The *fixed costs*. These are costs that remain the same, no matter how many items you produce. For example, if you are printing posters, the cost of the printing press is a fixed cost.
- The *cost per item*. This is how much it costs you to make one item. In our example, the cost of the ink and paper for one poster would be the cost per item. (These are also called *variable costs*, as they depend on how many items you produce.)
- The *unit price*. This is how much you charge the customer for each item.

The formula for determining the break-even quantity is:

$$n = \frac{\text{fixed costs}}{\text{unit price} - \text{cost per item}}$$

Write a program called *BreakEven.java* that prompts the user for the fixed cost, cost per item, and unit price. The program will then calculate and display the minimum quantity required to break even.

Here is what the program might look like when you run it:

```
Enter fixed costs: $1000
Enter cost to produce each item: $.70
Enter unit price of each item: $1.10
Break-even quantity is 2500.
```

**Exercise 3.11** The goal of this exercise is to show the need for planning a program. The calculations themselves are not sophisticated, but figuring out *what* those calculations are requires some thinking.

Consider a staircase as shown in Figure 3.1.

Your job is to write a program that prompts the user for the number of steps in the staircase. It will then prompt for and read the width, rise, and run of a step in centimeters. The program will then calculate the number of cubic centimeters of concrete necessary to build the staircase. Round the number of cubic centimeters by adding 0.5 to the calculated volume and then casting that result to an integer. In general,

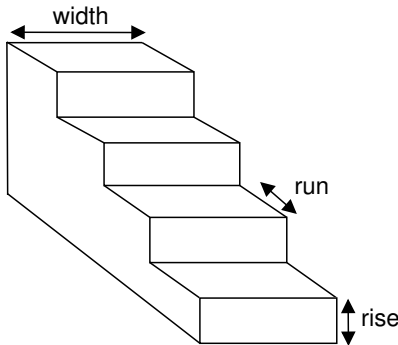


Figure 3.1: A staircase showing the width, the *rise* (height), and the *run* (depth) of a step.

```
double x;
int xRounded = (int) (x + 0.5);
```

Calculate by hand what the result of this code would be when `x` is 3.4 and again when `x` is 3.6 to understand why this code works.

*Hints:* How many “blocks” the size of the bottom step do you need to build the staircase? Drawing a diagram *really* helps. If you have  $n$  steps, the total number of blocks you need is the sum of the numbers 1 through  $n$ , which you can calculate with this formula:

$$blocks = \frac{n(n+1)}{2}$$

Here is what program output might look like:

```
Staircase Volume Calculator
How many steps in the staircase? 5
Enter step width in cm: 45
Enter run of the step in cm: 20
Enter rise of the step in cm: 7.5
Total volume is 101250 cubic centimeters.
```