Attaching the opening parenthesis to a function is something that almost every programmer does, but for the rest, styles of placing white spaces differ between programmers (my style of placing a space before the closing parenthesis is rare). You can choose your own style in this respect, you do not need to follow mine. But I recommend that you use your chosen style consistently, which will make your code more readable even for programmers who use a different style.

Note that in the code above there is a hash mark (#) on the first line, with a text after that which explains some details of the code. The line with the hash mark is a comment line: whenever you put a hash mark in your code (except when it is within a string, of course), everything to the right of the hash mark for the remainder of the line is commentary, which Python ignores. You can use comments to clarify your code, if such clarification is needed. More details on providing comments to code I will give in a later chapter.

What you learned

In this chapter, you learned about:

- Using the **`print()`** function to display results
- Data types string, integer, and float
- Calculations
- Basic string expressions
- Type casting between strings, integers, and floats, using **`str()`**, **`int()`**, and **`float()`**

Exercises

Exercise 3.1 The cover price of a book is $24.95, but bookstores get a 40 percent discount. Shipping costs $3 for the first copy and 75 cents for each additional copy. Calculate the total wholesale costs for 60 copies.

Exercise 3.2 Can you identify and explain the errors in the following lines of code? Correct them.

exercise0302.py

```
print( "A message" ).
print( "A message' )
print( 'A messagef"' )
```

Exercise 3.3 When something is wrong with your code, Python will raise errors. Often these will be "syntax errors" that signal that something is wrong with the form of your code (e.g., the code in the previous exercise raised a `SyntaxError`). There are also "runtime errors," which signal that your code was in itself formally correct, but that something went

wrong during the code's execution. A good example is the `ZeroDivisionError`, which indicates that you tried to divide a number by zero (which, as you may know, is not allowed). Try to make Python raise such a `ZeroDivisionError`.

Exercise 3.4 Here is another illustrative example of a runtime error. Run the follow code and study the error that it generates. Can you locate the problem?

exercise0304.py

```
print( ((2*3)/4 + (5-6/7)*8 )
print( ((12*13)/14 + (15-16)/17)*18 )
```

Exercise 3.5 You look at the clock and see that it is currently 14.00h. You set an alarm to go off 535 hours later. At what time will the alarm go off? Write a program that prints the answer. Hint: for the best solution, you will need the modulo operator.