 Answers to review questions from Chapter 4

1. What are the three standard file streams defined by the **<iostream>** library?

**cin, cout, and cerr**

2. What are the formal names for the **<<** and **>>** operators?

***insertion* and *extraction***

3. What value do the **<<** and **>>** operators return? Why is this value important?

**These operators return the stream argument by reference. This convention makes it possible to chain several input/output operations together in a single statement.**

4. What is a *manipulator?*

**A *manipulator* is a special value or function used to control formatting.**

5. What is the difference between a *transient* and a *persistent* property?

**A *transient property* applies only to the next value. A *persistent property* remains in effect for a stream until it is explicitly changed.**

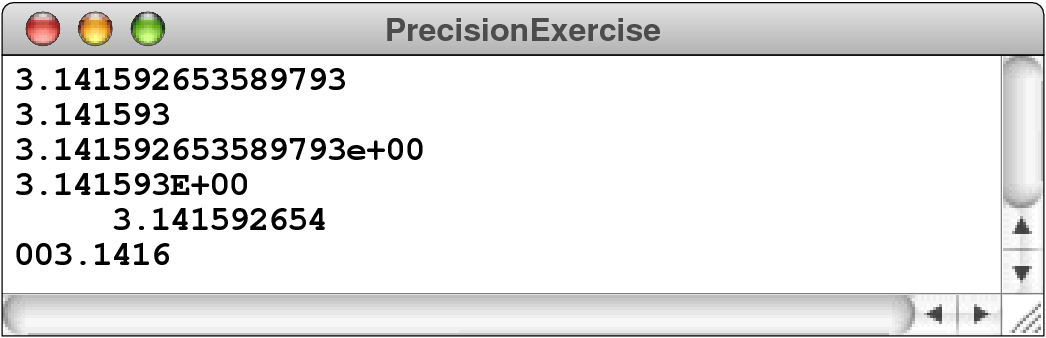
6. In your own words, describe how the **fixed** and **scientific** manipulators change the format for floating‑point output. What happens if you don’t specify either of these options?

**The fixed manipulator specifies that floating‑point values should always appear as a string of digits with a decimal point in the appropriate position. Conversely, the scientific manipulator specifies that values should always use the programming form of scientific notation in which the exponent is separated from the value by the letter E. If neither manipulator appears, floating‑point values are displayed in the most compact form.**

7. Suppose that the constant **PI** has been defined as

const double PI = 3.14159265358979323846;

What output manipulators would you use to produce each line of the following sample run:



cout << fixed << setprecision(15) << PI << endl;

cout << fixed << setprecision(6) << PI << endl;

cout << scientific << setprecision(15) << PI << endl;

cout << uppercase << scientific << setprecision(6) << PI << endl;

cout << fixed << setw(16) << setprecision(9) << PI << endl;

cout << fixed << setfill('0') << setw(8) << setprecision(4) << PI << endl;

8. What is the purpose of the types **ifstream** and **ofstream**?

**These types represent input and output file streams and are used to read and write data files.**

9. The argument to **open** must be a C‑style string. How does this requirement affect the code you write to open a file?

**If the name of the file is stored in a string variable, you need to apply the c\_str method to obtain the C‑style string.**

10. How can you determine if an **open** operation on a stream was successful?

**If the open method succeeds, calling the fail method returns false. If some error occurs, the fail method returns true.**

11. When you are using the **get** method to read a file character by character, how do you detect the end of a file?

**When it is used without an argument, the get method returns an integer value, which has the value EOF at the end of the file. The method get(ch) reads a character into the variable ch and then returns the stream, which can be used as a Boolean value specifying that no failure has occurred.**

12. Why is the return type of **get** declared as **int** instead of **char**?

**The get method returns an integer so that it is possible to return the special EOF sentinel, which has the value −1.**

13. What is the purpose of the **unget** method?

14. When you are using the **getline** method to read a file line by line, how do you detect the end of the file?

15. What classes does the **<sstream>** library support? How do these classes differ from the ones provided in **<fstream>**?

16. What is meant by the following terms: *subclass,* *superclass,* and *inheritance?*

17. True or false: The **stream** class hierarchy of Figure 4‑7 shows that **istream** is a subclass of **istringstream**.

18. Why does the **copyStream** function take arguments of type **istream** and **ostream** instead of **ifstream** and **ofstream**?

19. What are the advantages of using the **getInteger** and **getReal** functions from **simpio.h** over using the **>>** extraction operator?

20. If this text does not describe the functions exported by a library in tabular form, what options do you have for learning how to use that library?