Chapter 7

Text I/O

OBJECTIVES

After studying this chapter you will be able to:

- **Understand the basic design and operation of text files.**
- **Understand the C++ file implementation (streams).**
- **Name and use the four C++ standard file streams.**
- **Open and close application files.**
- **Read, write, and append text files.**
- **Format output data for presentation to the user.**
- **Use character input/output functions to read or write files**
- **Validate data to ensure accuracy.**

INPUT AND OUTPUT ENTITIES

Note:

A file on a disk or tape can be a text file or a binary file.

Note:

A file on a disk or tape can be a text file or a binary file.

Note:

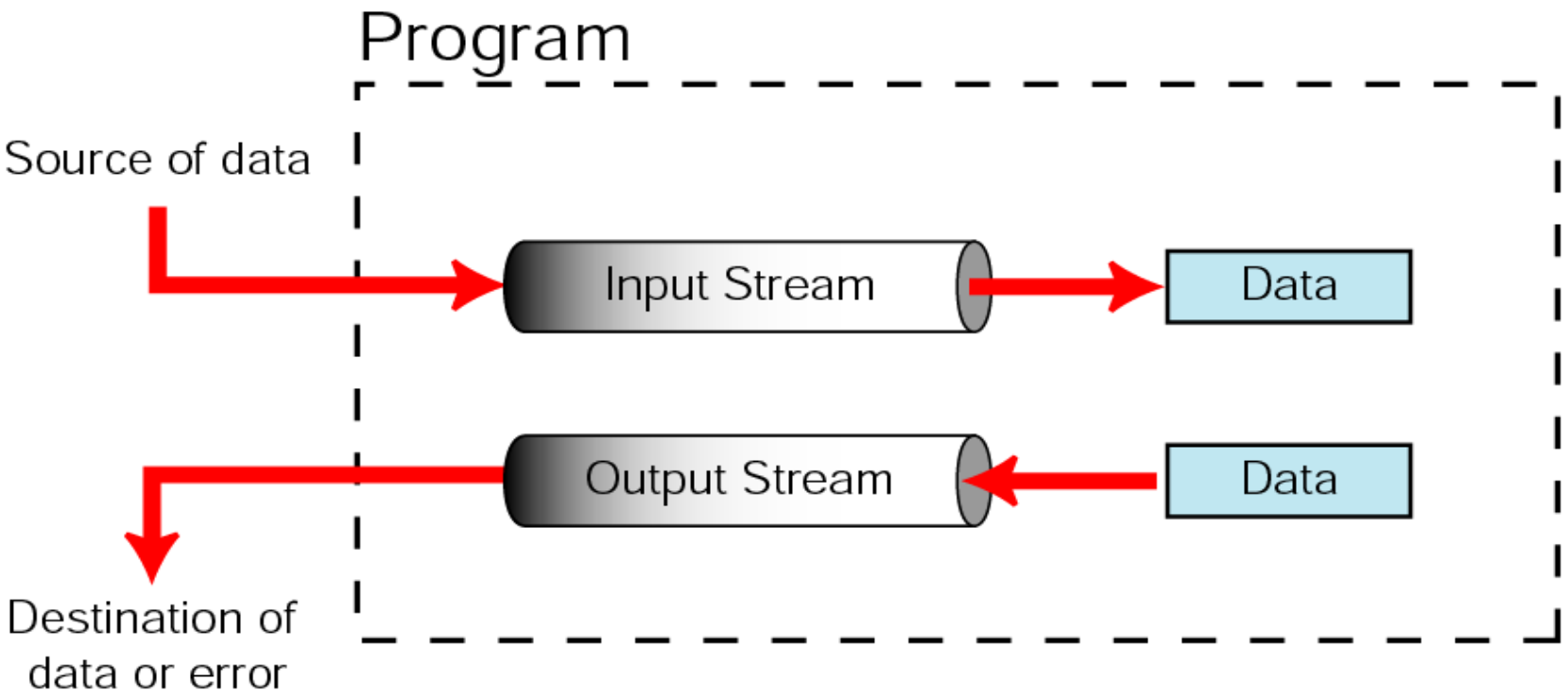
The standard input and output behave like text files.

Note:

The standard error behaves like a text file.

STREAMS

Figure 7-1 The stream concept



Note:

Standard streams are created, connected, and disconnected automatically.

Figure 7-2 Standard streams

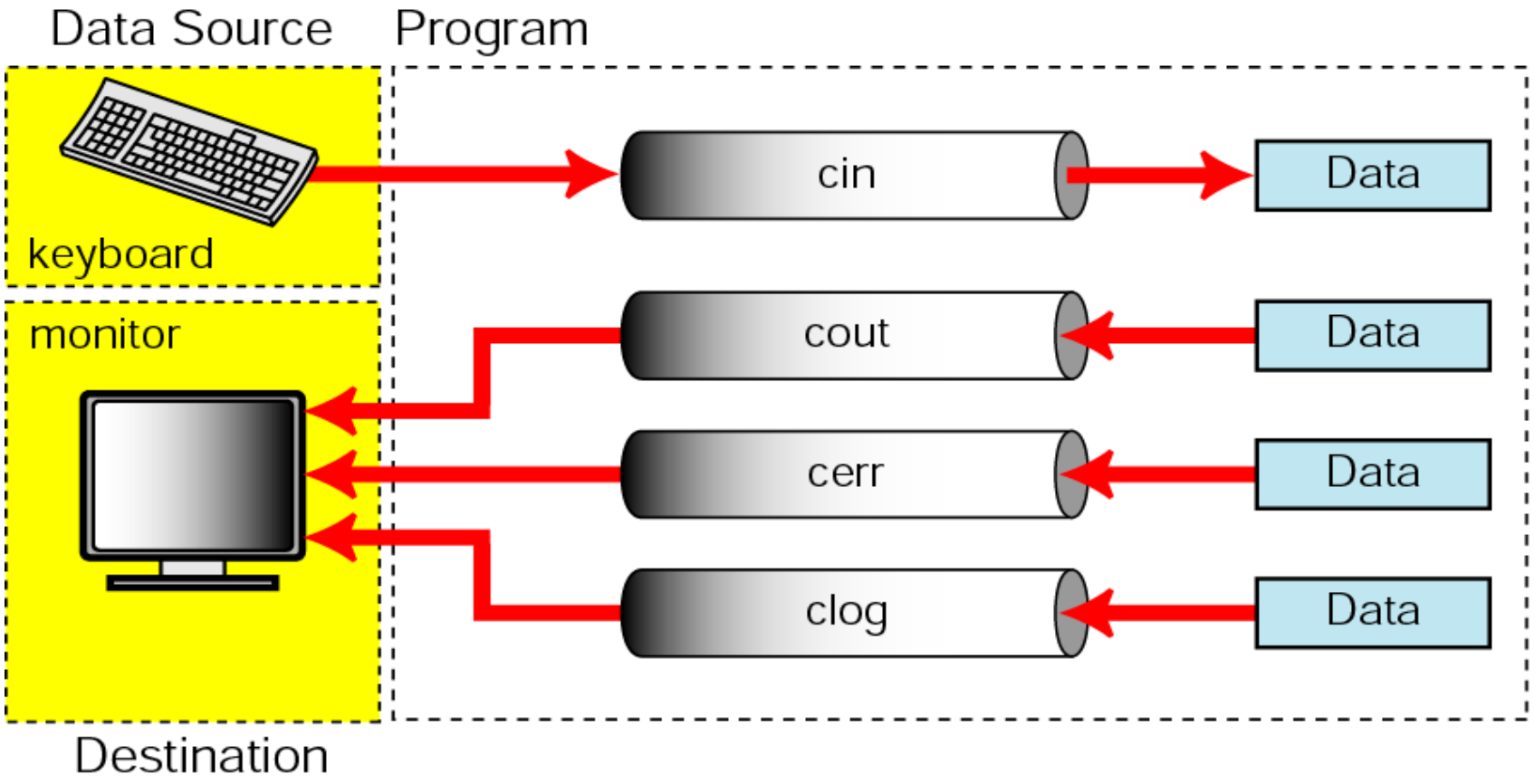
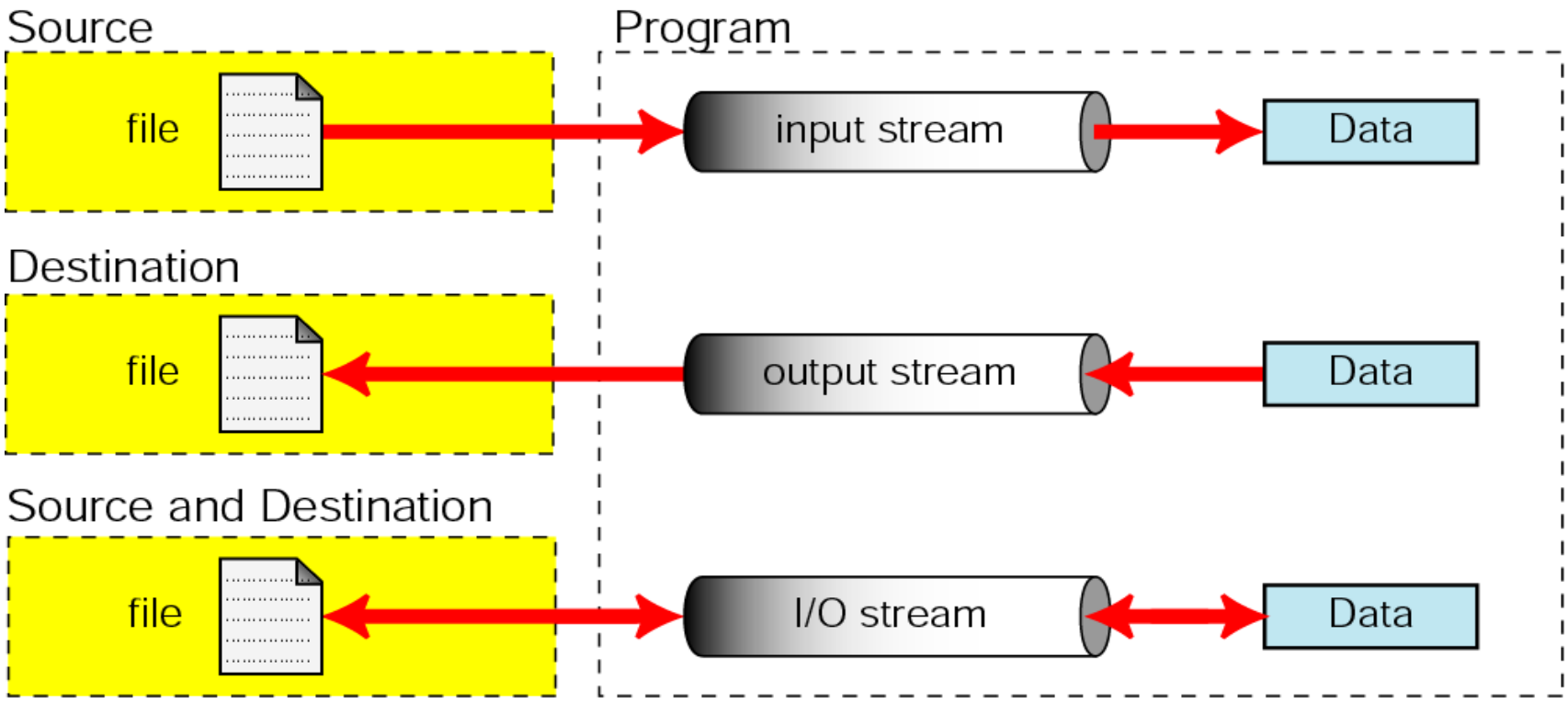


Figure 7-3 Connected file streams



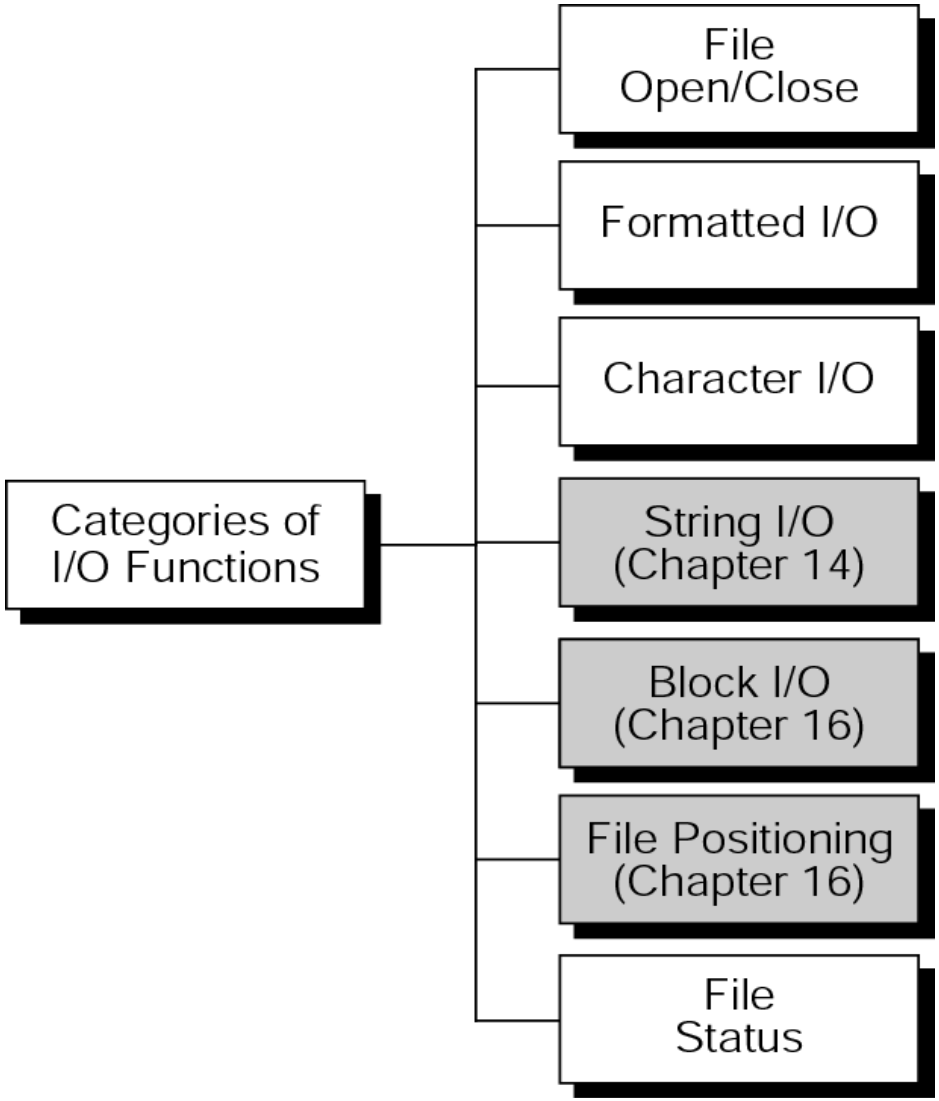
Note:

File streams are created, connected to files, and disconnected from files by the programmer.

7.3

Standard Library Input/Output Functions

Figure 7-4 Types of standard input/output functions

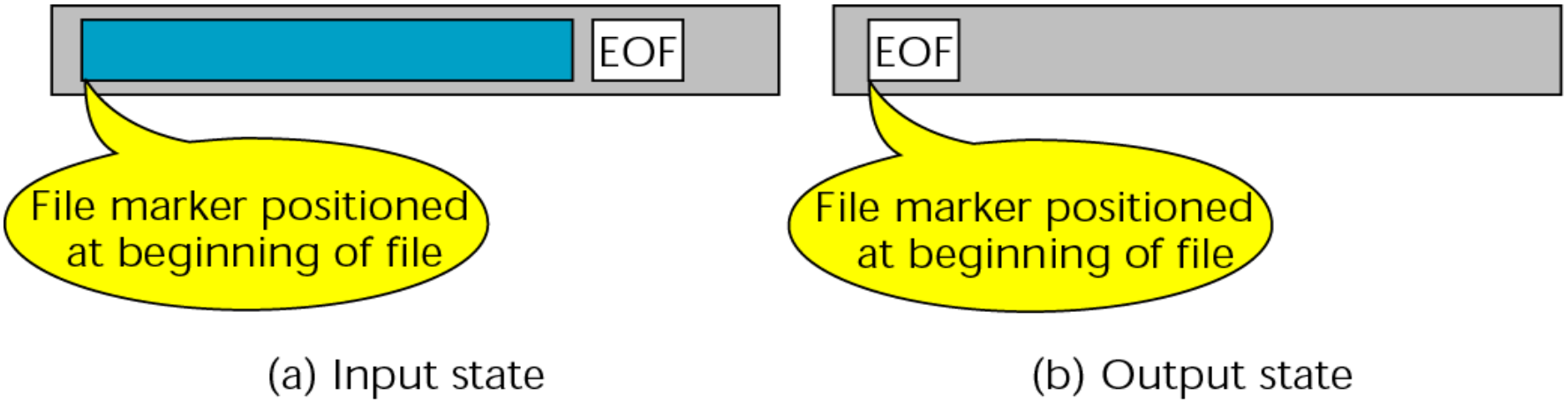


Note:

File Disposition Warning

The operating system controls what happens when you open an existing file for writing. The one consistent thing is that a new file is created. If you are in a UNIX or an MS/DOS environment, the existing file is deleted when the program completes. In other environments, the existing file still exists, but it is no longer the current file. To read it, you would have to use special job control statements that refer to the older version. Check with the documentation for your operating system to make sure you understand what will happen.

Figure 7-5 File opening states



(a) Input state

(b) Output state

7.4

Formatting Input and Output

Note:

Always use the adjustfield flag with the left and right flags.

Note:

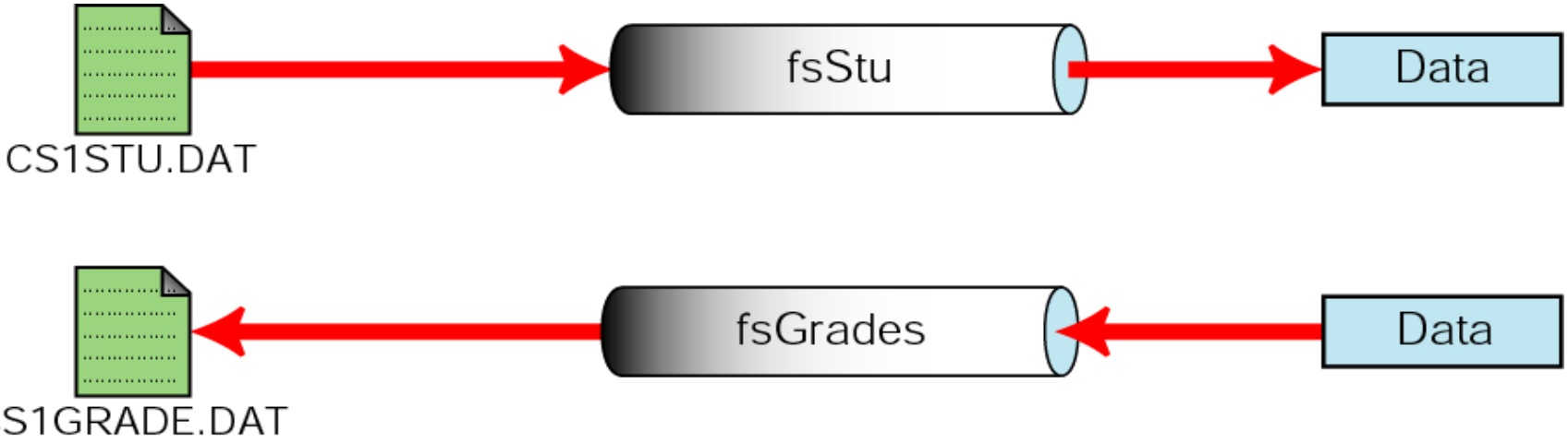
Always use the basefield flag with the dec, oct, and hex flags.

Note:

Always use the floatfield flag with the fixed and scientific flags.

FILE EXAMPLE

Figure 7-6 Create student grades




```
ifstream    fsStu;
ofstream    fsGrades;
```

CHARACTER INPUT/OUTPUT FUNCTIONS

CHARACTER INPUT/OUTPUT EXAMPLES

DETECTING FILE ERRORS

Software Engineering and Programming Style