**658** Chapter 12 Advanced File Operations

on the fstream data type. As a review, Table 12-1 compares the ifstream , ofstream , and

fstream data types. All of these data types require the fstream header file.

**Table 12-1** File Stream

**Data Type Description**

ifstream Input File Stream. This data type can be used only to read data from files into

memory.

ofstream Output File Stream. This data type can be used to create files and write data to them.

fstream File Stream. This data type can be used to create files, write data to them, and read

data from them.

**Using the fstream Data Type**

You define an fstream object just as you define objects of other data types. The following

statement defines an fstream object named dataFile .

fstream dataFile;

As with ifstream and ofstream objects, you use an fstream object’s open function to

open a file. An fstream object’s open function requires two arguments, however. The first

argument is a string containing the name of the file. The second argument is a file access

flag that indicates the mode in which you wish to open the file. Here is an example.

dataFile.open("info.txt", ios::out);

The first argument in this function call is the name of the file, info.txt . The second argument

is the file access flag ios::out . This tells C++ to open the file in output mode. Output

mode allows data to be written to a file. The following statement uses the ios::in access

flag to open a file in input mode, which allows data to be read from the file.

dataFile.open("info.txt", ios::in);

There are many file access flags, as listed in Table 12-2 .

**Table 12-2**

**File Access Flag Meaning**

ios::app Append mode. If the file already exists, its contents are preserved and all

output is written to the end of the file. By default, this flag causes the file to

be created if it does not exist.

ios::ate If the file already exists, the program goes directly to the end of it. Output

may be written anywhere in the file.

ios::binary Binary mode. When a file is opened in binary mode, data are written to or

read from it in pure binary format. (The default mode is text.)

ios::in Input mode. Data will be read from the file. If the file does not exist, it will

not be created and the open function will fail.

ios::out Output mode. Data will be written to the file. By default, the file’s contents

will be deleted if it already exists.

ios::trunc If the file already exists, its contents will be deleted (truncated). This is the

default mode used by ios::out .