C++ Programming:
From Problem Analysis to Program Design, Fourth Edition

Chapter 3: FileInputOutput

Objectives

In this chapter, you will:

- Learn how to obtain data from other input devices, such as a disk (that is, secondary storage), and how to save the output to a disk.

File Input/Output

Inputting data in a program from the keyboard is comfortable as long as the **amount of input is very small**.

If the **amount of input data is large**, however, it is inefficient to type it at the keyboard each time you run a program.

In addition to the inconvenience of typing large amounts of data, typing can generate errors, and unintentional typos cause erroneous results.

File Input/Output

File: An area in secondary storage used to hold information.

The standard I/O header file, **iostream**, contains data types and variables that are used only for input from the standard input device and output to the standard output device.

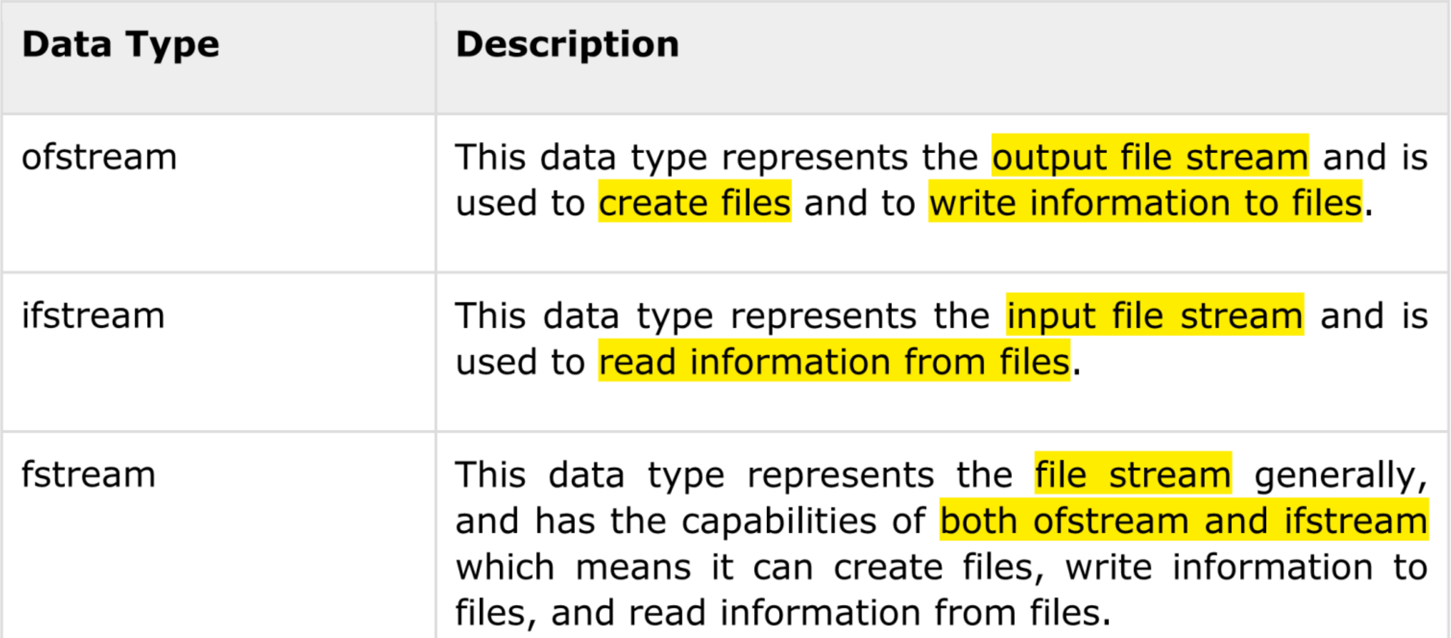
The fstream header file contains the definitions of two data types: ifstream, which means input file stream and is similar to istream, and ofstream, which means output file stream and is similar to ostream.

File Input/Output

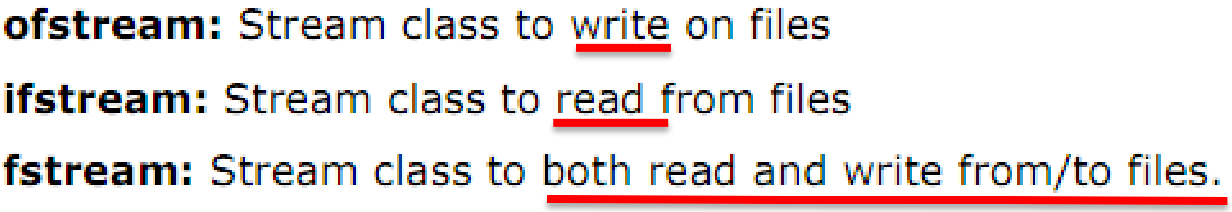
File I/O is a five-step process:

1. Include the header file fstream in the program.
2. Declare file stream variables.
3. Associate the file stream variables with the input/output sources. This step is called opening the files.
4. Use the file stream variables with >>, <<, or other input/output functions.
5. Close the files.

File Input/Output



Data Type	Description
ofstream	This data type represents the output file stream and is used to create files and to write information to files.
ifstream	This data type represents the input file stream and is used to read information from files.
fstream	This data type represents the file stream generally, and has the capabilities of both ofstream and ifstream which means it can create files, write information to files, and read information from files.

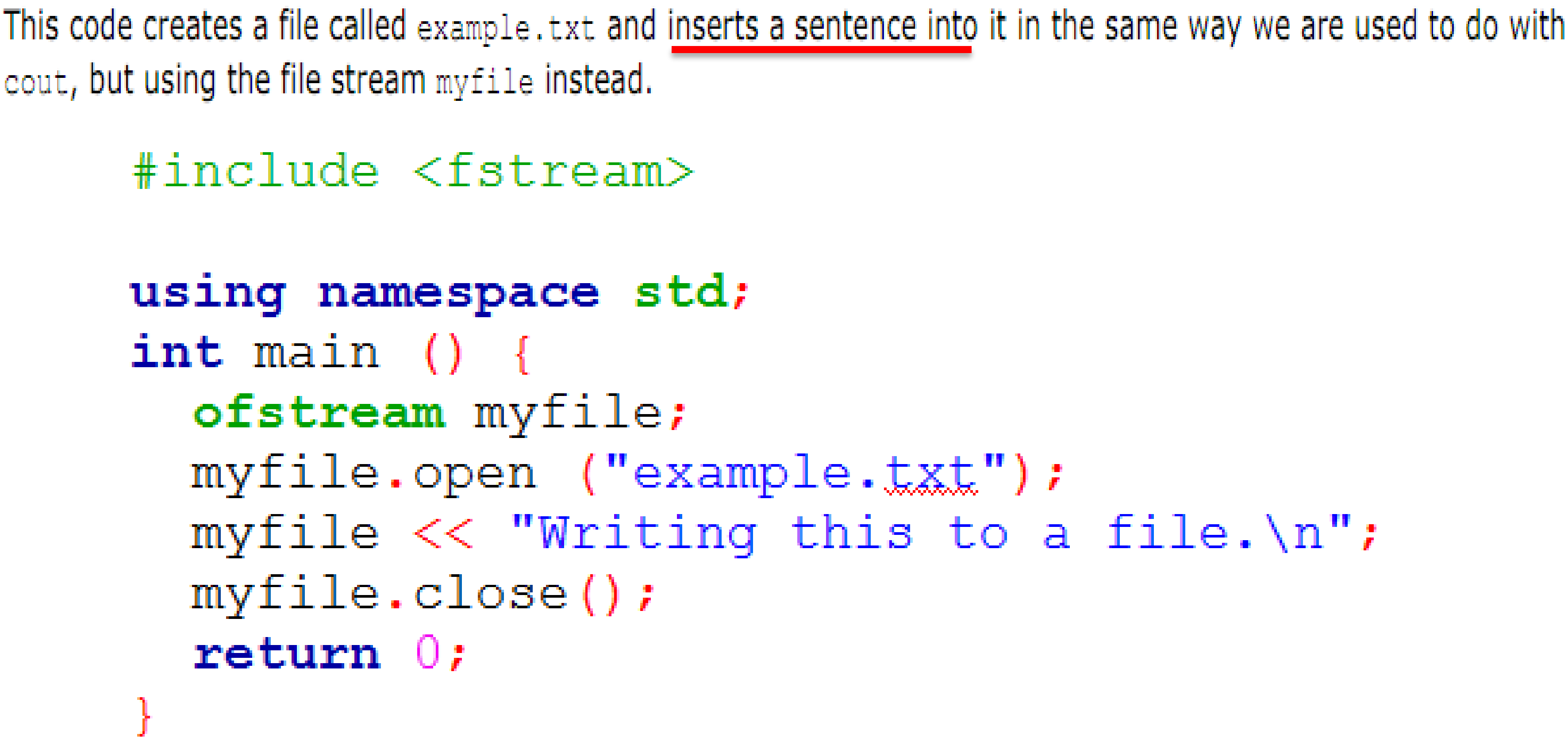


ofstream: Stream class to write on files

ifstream: Stream class to read from files

fstream: Stream class to both read and write from/to files.

File Input/Output



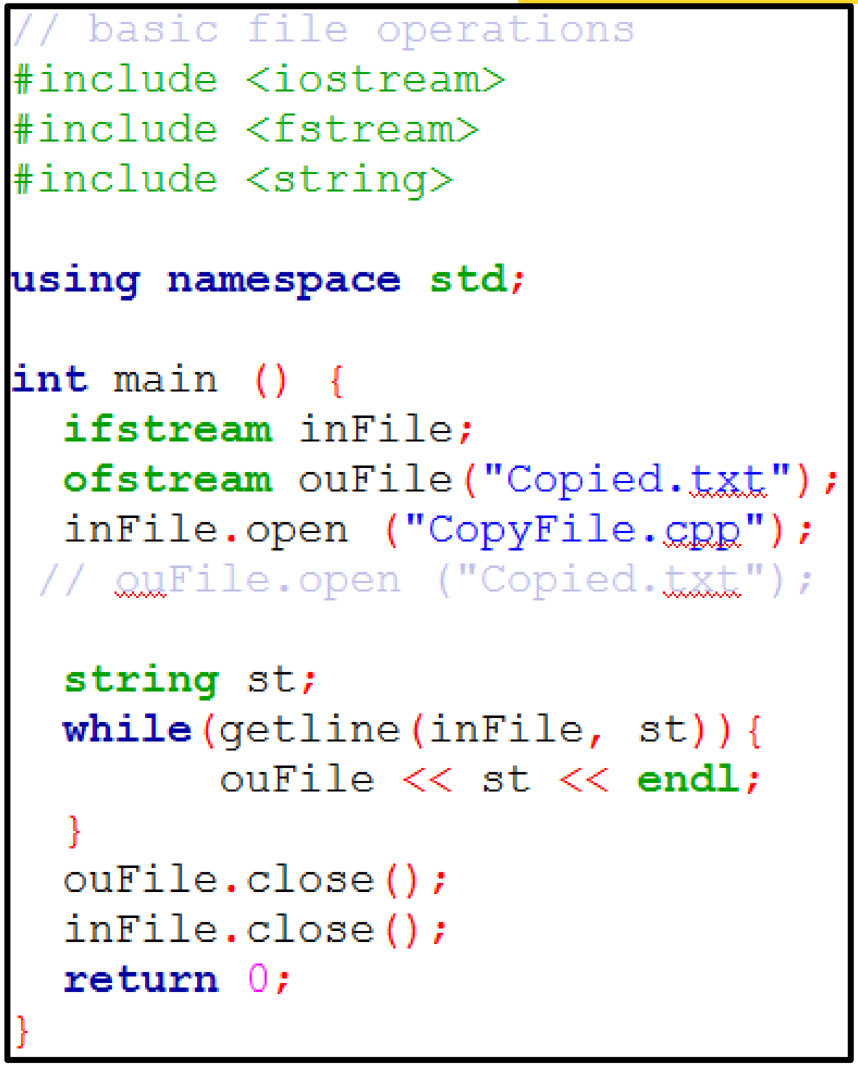
This code creates a file called `example.txt` and inserts a sentence into it in the same way we are used to do with `cout`, but using the file stream `myfile` instead.

```
#include <fstream>

using namespace std;
int main () {
  ofstream myfile;
  myfile.open ("example.txt");
  myfile << "Writing this to a file.\n";
  myfile.close();
  return 0;
}
```

Example: Copy one file to another

getline reads in the characters of each line until it discovers a newline.

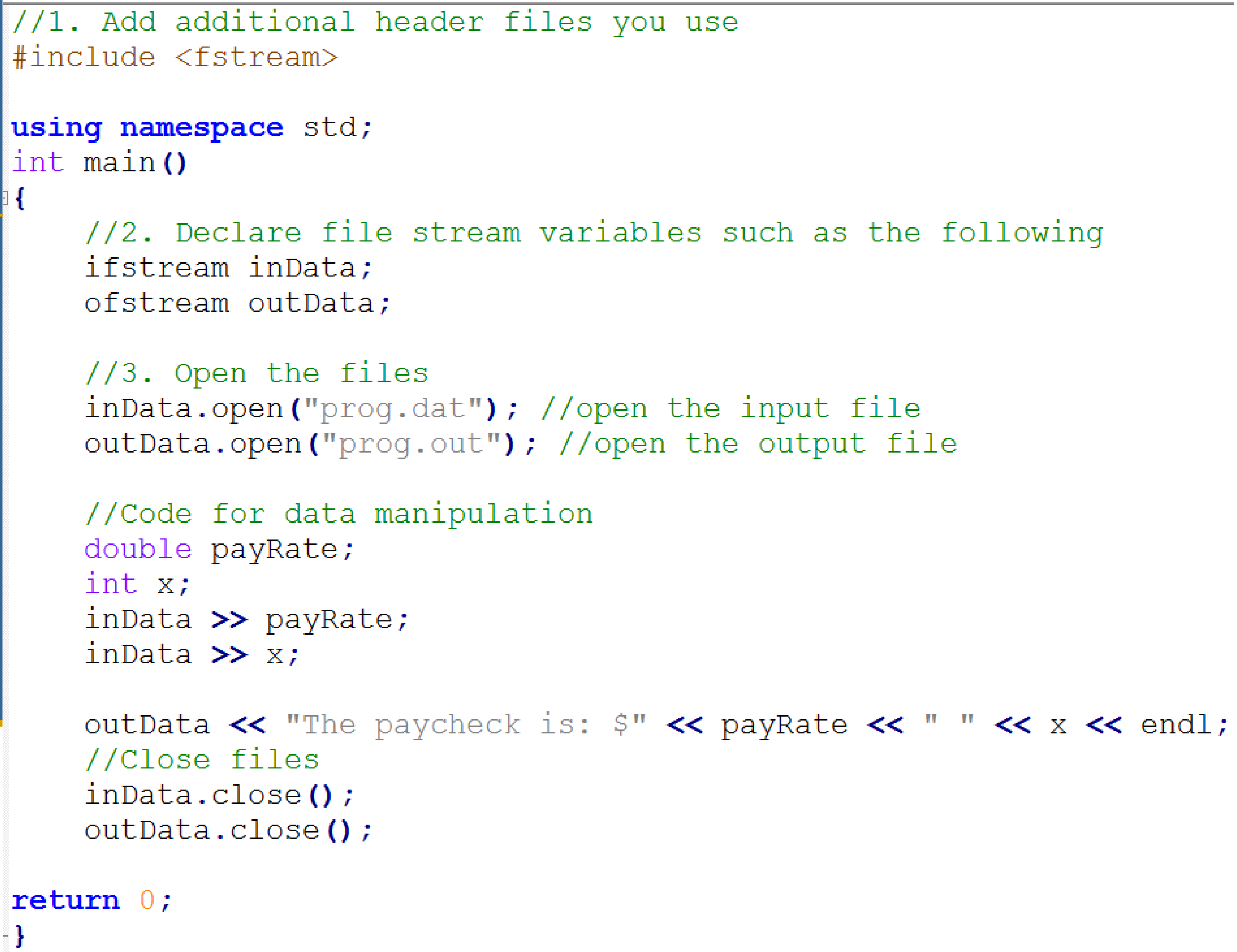
getline(ifstream in, string st) will return a value that can be interpreted as "true" if another line has been read successfully, and "false" upon reaching the end of the input.



```
// basic file operations
#include <iostream>
#include <fstream>
#include <string>

using namespace std;

int main () {
  ifstream inFile;
  ofstream ouFile("Copied.txt");
  inFile.open ("CopyFile.cpp");
 // ouFile.open ("Copied.txt");

  string st;
  while(getline(inFile, st)){
        ouFile << st << endl;
  }
  ouFile.close();
  inFile.close();
  return 0;
}
```

Problems

1. Write a program to copy the entire file into a single string object.
2. Write a program to read a text file using eof() function.
3. Create a program that opens a file and counts the whitespace-separated words in that file.
4. Create a program that counts the occurrence of a particular word in a file (use the stringclass' operator '==' to find the word).
5. Display a file a line at a time, waiting for the user to press the "Enter" key after each line.



```cpp
//1. Add additional header files you use
#include <fstream>

using namespace std;
int main()
{
    //2. Declare file stream variables such as the following
    ifstream inData;
    ofstream outData;

    //3. Open the files
    inData.open("prog.dat"); //open the input file
    outData.open("prog.out"); //open the output file

    //Code for data manipulation
    double payRate;
    int x;
    inData >> payRate;
    inData >> x;

    outData << "The paycheck is: $" << payRate << " " << x << endl;
    //Close files
    inData.close();
    outData.close();

return 0;
}
```