File I/O with Member Functions

Yash Raj,
Department of CSE, SOET
Central University of Haryana,Mahendragarh

202134

Abstract

The file is a collection of any kind of data that is stored on the disk. Anything written inside the file is called a " patent ", for example, "int" is a patent, and" Elephant" is a patent. The text file is the combination of multiple types of characters, numbers, and some special characters, for example, the semicolon ";" is a character, and"@" is a special character. The computer read these characters in the file with the help of some special unique code, the ASCII code. Each and every character is mapped on some decimal number (0 to 9). As for example, the ascii code for the character small letter "c" is "99" similarly for character "b" is 98 which is a decimal number. These decimal numbers are converted into a binary numbers (0 and 1) to make them readable for the computer system because the computer (CPU) can only understand the binary language "0" and "1".

The reason that computers can only understand binary numbers is that a computer is made up of millions of small switches (transistors) and switches only perform two operations "on" or "off" or "true" or "false" or "1" or "0".

The file can be of any type whether it is a file of programs like a ".js" file, ".cpp",".exe" file, a file of a game, or any other type of file.

1. Input/output with files

Two main operations that can be often performed on the files in a file IO –

- The read operation and,
- The write operation.

The image below shows the file read and write method through the C++ program.

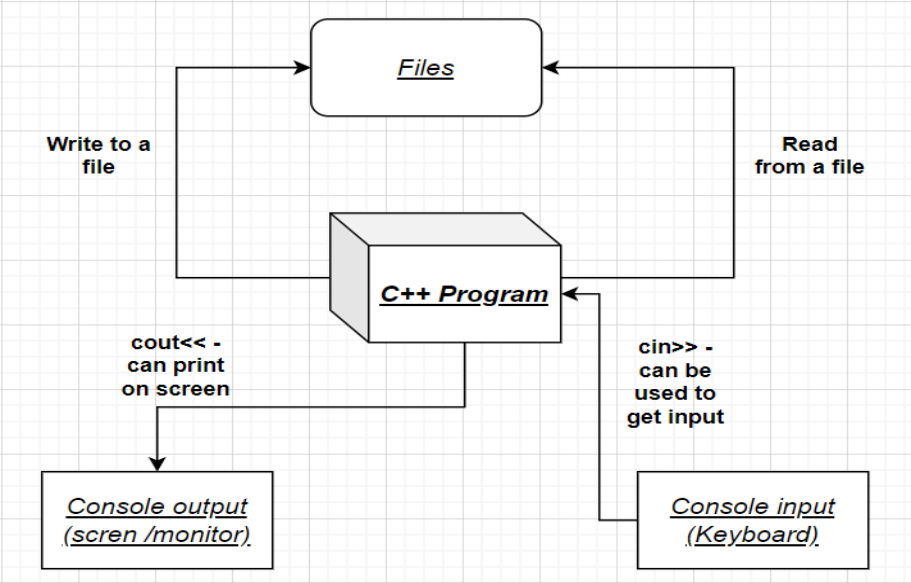

Figure 1: File IO Read and Write Diagram.
As shown in figure 1,

- The user can give any input to the C++ program by using the keyboard through the “cin” keyword along with the “>>” operator.
- The user can get the output from the C++ program on the monitor screen through the “cout” keyword along with the”<<” operator.
- The user can write on the file which is saved on the disk.
- The user can read the file which is saved on the disk.

2. File I/O using C++ - Reading and Writing from and to the Files.

C++ programming language dispense some of the classes to execute and process the output and input processes to and from the files that is- the file operations.

- ofstream class: The stream class to write on the files.
- ifstream class: The stream class to read from the files.
- fstream class: The stream class to both read and write from and to the files.

These classes are directly or indirectly derived from the istream and ostream classes. These objects were already in use in the early days of programming in C ++ whose kinds were these classes for example cout is an object of class ostream . We previously have been using the classes that are associated to our file-streams. And in fact, we can use our file streams in the exact way we are used to use cin and cout objects with the only alteration being that we have to subordinate these streams through files and operations in order to use them and work on a file using a C++ programming language.

2.1 Opening a File – for reading and writing operations

To work with files in C++, or in fact, in any programming language, we will have to open it at first. Now, there are 2 ways to open a file (particularly in C/ C++) :

- Using the constructor of ofstream or ifstream or iostream class.

- Using the member function open( ) of the class fstream
  Example :

```
// opening and Writing to a file…
#include <iostream>
#include <fstream>   //fstream header file is included…
using namespace std;

int main (void) {
  ofstream obj;   //ofstream class object is created.
  obj.open ("sample.txt");    //file is opened using member function open( ).
  obj << "Writing to a file.\n";
  obj.close( );  //finally a file is closed using close( ) member  function
  return 0;
}
```

The code above creates a simple .txt file named “sample.txt” and inserts a passed sentence into it in the file in a similar way we use with cout using << operator, but using the file stream ‘obj’ object instead. In the same way, we can read the contents of a file sample.txt using the >> operator in the same way we use cin, but using the file class stream ‘obj’ object instead.

OPENING A FILE :

1. Using the constructor :
   Syntax:
   ofstream file_object(“file_name.txt”, mode);

2. Using the open( ) member function :
   Syntax:
   ofstream file_object;
   file_object.open(“file_name.txt”,mode);

In both cases there is a ‘mode’, an optional parameter that specifies the mode to open a file i.e. whether we want to open it only for reading, writing (or both), appending, truncating, etc. We can open a file in C++ with a combination of some flags:

Table 1.

flags	**specification**
ios::in	open a file for reading operations.
ios::out	open a file for writing operations.
ios::binary	open a file in binary mode.
ios::app	open a file in append mode.
ios::trunc	If the file is opened for output operation and it already exists, then the content that already exists previously gets deleted and replaced by the new content.

Note: These flags can be used altogether by using the bitwise operator OR (|).

Note: Every open member function of class ofstream, ifstream, and fstream has a default mode that is used if the file is opened without a second argument i.e. the second argument is optional. For ofstream class the default mode parameter is (ios::out), similarly for ifstream default mode is (ios::in) and therefore for fstream default is (ios::out | ios::in)

Example: with reference to Table 1.

```
ofstream file ("sample.dat", ios::out | ios::app | ios::trunc);
```

Alternatively,

```
ofstream file;
file.open("sample.dat",ios::out|ios::app|ios::trunc);
```

2.2 Closing a File

Once we are done with our work on a particular file we shall always close that file so that it can be opened again and we can perform the operations over, for this , we call a member function close( ) to close any file.

Syntax: file.close( );

It is a good practice to use the close ( ) member function to close the file.

3. File I/O in C++ : Member functions

As we know C++ file I/O classes such as fstream,ifstream, and,ofstream are predefined classes that are developed to work or handle the operations on file(s) which are stored in a disk.
These classes have some predefined member functions which provide very useful utility tools for file operations.
In C++ programming language, there are different types of public member functions related to different file input-output operations.

Some of the Public Member functions for file operations –

- (constructor) - constructor object and optionally open file.
- open( ) - open file.
- is_open( ) - check if a file is open or not.
- close( ) – close a file.
- Swap( ) – swaps internals.
- rdbuf( ) – get the connected filebuf object.
- operator= - move assignment.

Some public member functions inherited from **istream** –

- operator>> - Extract formatted input.
- gcount( ) - get character count.
- get( ) - get a character.
- getline( ) - get a complete line as input.
- ignore( ) - extract and discard the characters.
- peek( ) - peek next character.
- read( ) - read the block of data .
- readsome( ) - read the data which is available in a buffer
- putback( ) – put character back.
- unget( ) - unget a character.
- tellg( ) - get the position in input sequence.
- sync( ) - synchronize input buffer.

Some public member functions inherited from **ostream** –

- operator<< - Insert formatted output.
- put( ) - put a character.
- write( ) - write a block of data.
- tellp( ) - get position in output sequence.
- seekp( ) - set position in output sequence.
- flush( ) - flush the output stream buffer.

Some public member functions inherited from **ios** –

- good( ) - check whether state of stream is good.
- eof( ) - check whether eofbit is set.
- fail( ) - check whether either failbit or badbit is set.
- bad( ) - check whether badbit is set.
- rdstate( ) - Get error stage flags.
- rdbuf( ) - get/set stream buffer.

4. Some Useful Member functions in File I/O :

4.1. open( ) :

In order to work with file in c++ we first have to open the file and we already know there two ways to open a file i.e. using constructor or using open( ).
open( ) associate function associated with stream object:

readFile.open(“sample.txt”);
writeFile.open(“sample.txt);
This groups up the file streams to read the data from a file named “sample.txt” and write the output to the same file “sample.txt”.

4.2. getline( ) :

The getline( ) standard library function offers a humble way to read a character input into a string variable. It reads the entire line from the file or the user input until it finds a

newline character ("\n") or endl and stores it into a variable.
For example, let's suppose we have the following file input:

Yash Raj	202134	Programmer
Shubham	202135	Programmer

Now let us assume 'inFile' is associated to the input file mentioned above, the statement

```
getline (inFile, buffer);   //string buffer
```

would result in buffer having the value:

"Yash Raj 202134 Programmer"

getline ( ) function accepts 2 simple arguments.The first specifies input stream and the second a string variable.
It is also possible to call getline( ) function with three parameters. The first 2 mentioned above along with the 3rd specify the "stop" that is the character in which method getline( ) will break reading after the input stream.

4.3. fail( ) :

fail( ) function offers a method to inspect the position of the last action on the input stream. fail( ), if the past action failed it returns a true value a returns false if the operation was successful.

```
#include <fstream>
using namespace std;
 void main( ) {
        ifstream instream;
        instream.open( "infile.dat");

        if ( instream.fail( ) ) {
                cout<<"File not Found !";
                return;
        }
 //…omitted statements doing something useful…
 instream.close( );
 }
```

4.4. get( ) :

The input stream object i.e. "cin" has a method named get( ) which returns the subsequent solo character in the stream, whether it is whitespace or not.
To invoke a method of an object, provide the name of an object, followed by the period, followed by a function call.

```
cin.get( SomeCharacter );   //SomeCharacter is some character variable.
```

get( ) function will eliminate the succeeding character from the stream 'cin' and place it in the variable SomeCharacter.

So in order to read some characters we can always have :

cin.get(SomeCharacter);

4.5. eof( ) :

Each file ends with a distinct character, called the end-of-file or EOF spot.
eof( ) is a Boolean function that returns 1 or true if the last input process attempted to read the eof spot, and returns 0 or false otherwise.
For example-

```
if ( !In.eof( ) )
        {
                out<<"An error occurred while reading a file."<<endl;
                out <<"Please check the input file"<<endl;
        }
....
```

4.6. peek( ) :

istream class offers several additional methods . The two important functions that are repeatedly used in file i/o operations are peek( ) and putback( )

Peek( ) offers a technique to inspect the subsequent character in the input stream, without eliminating it from the stream.
For instance:

```
....
char nextChar;
nextChar = In.peek( );
....
```

4.7. putback( ) :

putback( ) offers a method to return the latter character which is read to the input stream.
For instance:

```
....
const char PUTMEBACK = '?';
char nextChar;
In.get( nextChar);
if ( nextChar == PUTMEBACK ) {
        In.putback( nextChar);
}
....
```

Let us see some of the example code that uses the above-mentioned member functions
Example 1:

```
// Example of opening/creating a file using the open() function
#include <iostream>
#include <fstream>
using namespace std;
int main(void)
{
   fstream yash;
   yash.open("abc.dat", ios::out);
   if (!yash)
   {
      cout << "\nfile creation failed";
   }
   else
   {
      cout << "\nnew file is  created !!";
      yash.close( ); // Closing a file
   }
   return 0;
}
```

Example 2:

```
//A example program to close a file.
#include <iostream>
#include <fstream>
using namespace std;
int main(void)
{
fstream obj;
obj.open("new_file.txt",ios::out);
obj.close( );
return 0;
}
```

Example 3:

```
//Use of eof( ) and getline( ) functions.
#include <iostream>
#include<fstream>
using namespace std;
int main(void)
{
   ofstream yash("data.txt");
   char name[20];
   float cost;

   cout<<"Enter Name : ";
   cin>>name;
   cout<<"Enter cost : ";
   cin>>cost;

   yash<<"Name is "<<name;
   yash<<"\ncost = "<<cost;
   yash.close( );

   ifstream raj("data.txt");
   string container;
   while(raj.eof( )==0)
   {
      getline(raj,container);
      cout<<container<<endl;
   }
   raj.close( );
   return 0;
}
```

5. Interview Questions & Answers :

1. Which header file is required for file I/O operations in C++?
(https://www.sanfoundry.com/interview-questions-cplusplus-file-string-streams/)
Ans : <fstream>
2. Which of the following flag cannot be used as a file opening mode?
(https://www.sanfoundry.com/interview-questions-cplusplus-file-string-streams/)
a) ios::trunc
b) ios::binary
c) ios::in
d) ios::ate
Ans : a) ios::trunc
Explanation: If the file is opened for output operation and it already exists, then the content that already exists previously gets deleted and replaced by the new content. It is not a file-opening mode anyway.
(https://www.sanfoundry.com/interview-questions-cplusplus-file-string-streams/)

3. By default, In what mode all the files in C++ are opened?
Ans: By default, all the files in C++ for file I/O operations are opened in ".txt" -Text mode.

4. What is the use of ios::trunc mode?
(https://www.sanfoundry.com/interview-questions-cplusplus-file-string-streams/)
Ans: In C++ file handling, ios::trunc flag mode is used to truncate an existing file which means if the file is opened for output operation and it already exists, then its content that already exists previously gets deleted and replaced by the new content.
(https://www.sanfoundry.com/interview-questions-cplusplus-file-string-streams/)

5. What is the return type open( ) method?
a) int
b) char
c) bool
d) float
Ans: c) bool
Explanation: open( ) method returns a bool value indicating whether the file is opened or some error has occurred.
(https://www.sanfoundry.com/interview-questions-cplusplus-file-string-streams/)

6. Which function is used in C++ to get the current position of the file pointer in a file?
Ans: tell_p( )
Explanation: C++ provides tellp( ) function to get the current position of the file pointer in a file IO operations.
(https://www.sanfoundry.com/interview-questions-cplusplus-file-string-streams/)

7. Which member function in file I/O operation is used to position back the file pointer from the end of the file object?

Ans: seekg( )
Explanation: The member function seekg( ) is used to position back the file pointer from the end of the file object in the file io operation.
(https://www.sanfoundry.com/interview-questions-cplusplus-file-string-streams/)

8. Which file I/O member function is used to check whether the stream object is currently connected to a file?
Ans: is_open( )
(https://www.sanfoundry.com/interview-questions-cplusplus-file-string-streams/)

9. How many objects can be used for input and output to a string in C++?
Ans: 3
Explanation: The stringstream, ostringstream, and istringstream objects are used for input and output operations in a string.
(https://www.sanfoundry.com/interview-questions-cplusplus-file-string-streams/)

10. What is the difference between ifstream and fstream_stream class in C++ file I/O?
Ans: ifstream is a File handling class that signifies the input file stream and is used for reading the data from the file whereas fstream is also a file handling class that has the potential to handle both ifstream and ofstream. i.e it can be used to read from and write to a file both.

6. Problem statements

1. Determine the correct output of the given c++ code?

```
#include <iostream>
#include <fstream>
 using namespace std;
 int main (void)
 {
    int size;
    char * storage;
    ifstream ist;
    ist.open ("sample.txt", ios :: binary );
    ist.seekg (0, ios :: end);
    length = is.tellg( );
    ist.seekg (0, ios :: beg);
    buffer = new char [size];
    ist.read (storage, size);
    ist.close( );
    cout write (storage, size);
    delete[] storage;
    return 0;
 }
```

2. Find the output of the subsequent c++ code?

```
#include<iostream>
#include <fstream>
 using namespace std;
 int main (void)
   {
      ofstream yash ("sample.txt");
      for (int n = 1; n <= 100; n++)
      {
         yash << n;
         yash.flush( );
      }
      cout << "Dosti";
      yash.close( );
      return 0;
   }
```

(https://www.sanfoundry.com/interview-questions-cplusplus-file-string-streams/)

3. Write a c++ code to read (input) the name and roll number of students from the keyboard and write them into a file and then display it.

4. Develop a c++ Program to Count Digits Alphabets & Spaces using File I/O member functions.

5. Determine the output of the subsequent c++ code?

```
#include <iostream>
using namespace std;
int main (void)
        {
            char first, second;
            cout << "Enter a word: ";
            first = cin.get( );
            cin.sync( );
            second = cin.get( );
            cout << first << endl;
            cout << second << endl;
            return 0;
        }
```

(https://www.sanfoundry.com/interview-questions-cplusplus-file-string-streams/)

6. Develop a C++ program to Count the no. of words in a given file using File IO member functions.
7. Write a C++ code to explain the working of tellg( ) and tellp( ) member functions.

8. Which member function (or method) is used to re-locate the file pointer in C++ file I/O?
9. Find the correct output of the subsequent c++ code?

```
#include <iostream>
 using namespace std;
 int main (void)
  {
     char c;
     streambuf * ptr;
     ofstream obj ("sample1.txt");
     pbuf = obj.rdbuf( );
     do {
        c = cin.get( );
        ptr -> sputc(c);
     } while (c != '.');
     obj.close( );
     return 0;
  }
```

(https://www.sanfoundry.com/interview-questions-cplusplus-file-string-streams/)

10. Write a c++ code to take data from the user through the keyboard and write it on a file data.txt and read the file & show the file content.

7. Answers to the problem statements:

1. In this program, if the required file exists, it will read the file, otherwise, it will throw an exception. A runtime error will occur because the value of the size variable will be "-1" if the file 'sample.txt' does not exist and in line 13 we are trying to allocate an array of size "-1" since the value of variable size is -1.

2. Output: Dosti, since in this program, we are using the flush( ) function to update the contents in a file.

3.

```
#include <iostream>
#include<fstream>
using namespace std;

int main(void)
{
   ofstream infile;
   infile.open("studentData.txt",ios::out|ios::app);

   int rollno;   //variable to store roll number and name
   char name[20];

   cout<<"Enter Student's Name : ";     //Inputs from the user
   cin.getline(name,20);
   cout<<"Enter Student's Roll_no : ";
   cin>>rollno;


   //Writing into a file (studentData.txt)
   infile<<roll no<<" ";
   infile<<name<<"\n";
   cout<<"\nSucessfully written in file !\n";
   infile.close( );

   //Reading from a file (studentData.txt)
   ifstream outfile;
   outfile.open("studentData.txt",ios::in);
   string container;

   cout<<"\nFile content :\n";
   while(outfile.eof( )==0)
   {
      getline(outfile,container);
      cout<<container<<endl;
   }
   outfile.close( );
   return 0;
}
```

4.

```
/*C++ Program to count No. of Alphanumeric and No. of  Spaces using File IO
operations*/
#include <iostream>
#include<fstream>
using namespace std;
int main(void)
{
   ifstream fobj;
   infile.open("stdData.dat",ios::in);

   int digitCount=0;
   int alphaCount=0;
   int spaceCount=0;

   char ch;
   while (fobj.eof( )==0)
   {
      ch=fobj.get( );
      if(ch>63 && i<91 || ch>96 && ch<123)
      {
         alphaCount++;
      }
      else if (ch ==' ')
      {
         spaceCount++;
      }
      else if (ch>47 && ch<58)
      {
         digitCount++;
      }

   }

   cout<<"No of Alphabets = "<<alphaCount<<endl;
   cout<<"No of Blank spaces = "<<spaceCount<<endl;
   cout<<"No of Digits = "<<digitCount<<endl;
   fobj.close( );
   return 0;
}
```

5. The program will return the first 2 letters or numbers or any special character from the entered word by the user since we are using the sync( )
   For example, if the user enters Yash then the output is: Y
   a

6.

```
//C++ Program to count the No. of words in a given file using File IO functions
#include <iostream>
#include<fstream>
using namespace std;
int main(void)
{
   ifstream yash;
   yash.open("sample1.txt",ios::in);

   int count=0;
   char ch;
   while (yash.eof( )==0)
   {
      ch=yash.get( );
      if(ch==' '||ch=='\n')
      count++;

   }

   cout<<"No of words = "<<count+1<<endl;

   yash.close( );
   return 0;
}
```

7.

```
//A C++ program to demonstrate the use of tellg( ) and tellp( ) function.
#include <iostream>
#include <fstream>
using namespace std;
int main(void)
	{
		fstream yash;
		//open file data.txt in and Write mode
		yash.open("data.txt",ios::out);
		if(!yash)
		{
			cout<<"File cannot be created!!!";
			return 0;
		}
		//write random letters.
		yash<<"qwertyuiopasdfghjklzxcvbnm";
		//print the position
		cout<<"Current position is: "<<yash.tellp( )<<endl;
		yash.close( );
		//again open file in read mode
		yash.open("data.txt",ios::in);
		if(!yash)
		{
			cout<<"Error in opening file!!!";
			return 0;
		}
		cout<<"file position is(after opening): "<<yash.tellg( )<<endl;
		//read characters untill end of file is not found
		char ch;
		while(!yash.eof( ) )
			{
				cout<<"File pointer at position : "<<yash.tellg( );
				 //current position.
				yash>>ch;
				cout<<" Character \""<<ch<<"\""<<endl;
			}
		yash.close( );
		return 0;
	}
```

8. The member function seekg( ), In c++ programming language the member function seekg( ) is used to place a file pointer back from the EOF.

9. Dot operator(.). The program will eventually stop getting the inputs when it encounter dot operator.

10.

```
/*A c++ program to take data from the user through a keyboard and write it on a file
data.txt and read the file and display the file content.*/
#include <iostream>
#include<fstream>
using namespace std;
int main(void)
{
   ofstream outfile("data.txt");
   char name[20];
   float cost;

   cout<<"Enter Name : ";
   cin>>name;
   cout<<"Enter Cost : ";
   cin>>cost;

   outfile<<"Name is "<<name;
   outfile<<"\ncost = "<<cost;
   outfile.close( );

   ifstream inputfile("data.txt");
   string container;
   while(inputfile.eof( )==0)
   {
      getline(inputfile,container);
      cout<<container<<endl;
   }
   inputfile.close( );
   return 0;
}
```

8. Conclusion :

Up to now we have let the main ( ) function to handle all the particulars of file I/O. Once we use more classy classes it is natural or obvious to comprise file I/O operations as member functions of the class.

(Object-Oriented Programming in C++, Robert Lafore, p604)

9. References :

- (https://www.sanfoundry.com/interview-questions-cplusplus-file-string-streams/)

- (https://www.youtube.com/watch?v=cxFGaCoM4WE)
- (https://runestone.academy/ns/books/published/cpp4python/Input_and_Output/InputandOutput.html)
- (https://interviewmania.com/discussion/40709-cpp-programming-cpp-files-streams/)

- (Object-Oriented Programming in C++,Fourth Edition -Robert Lafore)
- (https://app.diagrams.net/ -for Figure 1)