

C++ Tutorial

Introduction to
C++
Programming
Language

Setting up C++
Development
Environment

C++
Programming
Basics

Functions in
C/C++

Arrays in
C/C++

Strings in C++
and How to
Create them?

→ Pointers in
C/C++ with
Examples

C++ Classes
and Objects

Object
Oriented
Programming
in C++

Namespace in
C++ | Set 1
(Introduction)

C/C++
Preprocessors

Templates in
C++

The C++
Standard
Template
Library (STL)

Exception
Handling in
C++

**File Handling
through C++
Classes**

new and
delete
operators in
C++ for
dynamic
memory

C++ | Signal
Handling



Multithreading
in C++

TAPE
Be a #FarmDost

Ad

100 FARMERS STORIES

100 Farmers. 100 Stories.
Photo & Video Story Contest

Participate and stand a chance to win prizes worth upto ₹2,00,000

Register Now

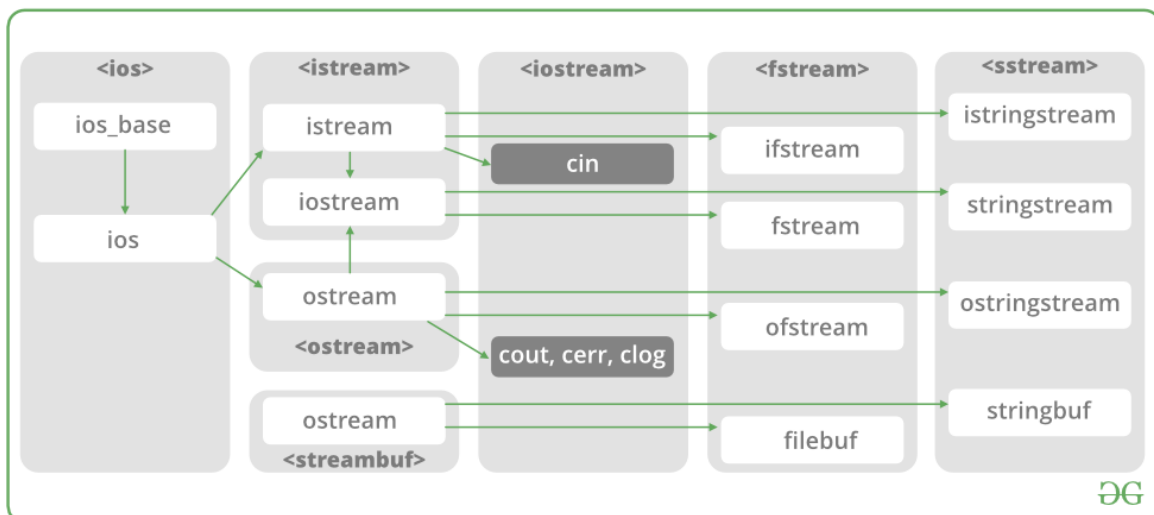
File Handling through C++ Classes

In C++, files are mainly dealt by using three classes `fstream`, `ifstream`, `ofstream` available in `fstream` headerfile.

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.



Now the first step to open the particular file for read or write operation. We can open file by

1. passing file name in constructor at the time of object creation

2. using the open method

For e.g.

Open File by using constructor

```
ifstream (const char* filename, ios_base::openmode mode = ios_base::in);
ifstream fin(filename, openmode) by default openmode = ios::in
ifstream fin("filename");
```

Open File by using open method

Calling of default constructor
 ifstream fin;



```
fin.open(filename, openmode)
fin.open("filename");
```

Modes :

MEMBER CONSTANT	STANDS FOR	ACCESS
in *	input	File open for reading: the internal stream buffer supports input operations.
out	output	File open for writing: the internal stream buffer supports output operations.

binary	binary	Operations are performed in binary mode rather than text.
ate	at end	The output position starts at the end of the file.
app	ap- pend	All output operations happen at the end of the file, appending to its existing contents.
trunc	trun- cate	Any contents that existed in the file before it is open are discarded.

Default Open Modes :

ifstream	ios::in
ofstream	ios::out
fstream	ios::in ios::out

Problem Statement : To read and write a File in C++.

Examples:

Input :

Welcome in GeeksforGeeks. Best way to learn things.

-1

Output :





Welcome in GeeksforGeeks. Best way to learn things.



Recommended: Please try your approach on {IDE} first, before moving on to the solution.

Below is the implementation by using **ifstream & ofstream** classes.

C++



```
/* File Handling with C++ using ifstream & ofstream class object*/
/* To write the Content in File*/
/* Then to read the content of file*/
#include <iostream>

/* fstream header file for ifstream, ofstream,
   fstream classes */
#include <fstream>

using namespace std;

// Driver Code
int main()
{
    // Creation of ofstream class object
    ofstream fout;

    string line;

    // by default ios::out mode, automatically deletes
    // the content of file. To append the content, open in ios::app
    // fout.open("sample.txt", ios::app)
    fout.open("sample.txt");

    // Execute a loop If file successfully opened
    while (fout) {

        // Read a Line from standard input
        getline(cin, line);

        // Press -1 to exit
        if (line == "-1")
            break;

        // Write line in file
        fout << line << endl;
    }

    // Close the File
    fout.close();

    // Creation of ifstream class object to read the file
    ifstream fin;

    // by default open mode = ios::in mode
    fin.open("sample.txt");

    // Execute a loop until EOF (End of File)
    while (fin) {

        // Read a Line from File
        getline(fin, line);





        // Print line in Console
        cout << line << endl;
    }

    // Close the file
```

Below is the implementation by using **fstream class**.

C++





```
/* File Handling with C++ using fstream class object */
/* To write the Content in File */
/* Then to read the content of file*/
#include <iostream>

/* fstream header file for ifstream, ofstream,
   fstream classes */
#include <fstream>

using namespace std;

// Driver Code
int main()
{
    // Creation of fstream class object
    fstream fio;

    string line;

    // by default openmode = ios::in|ios::out mode
    // Automatically overwrites the content of file, To append
    // the content, open in ios::app
    // fio.open("sample.txt", ios::in|ios::out|ios::app)
    // ios::trunc mode delete all content before open
    fio.open("sample.txt", ios::trunc | ios::out | ios::in);

    // Execute a loop If file successfully Opened
    while (fio) {

        // Read a Line from standard input
        getline(cin, line);

        // Press -1 to exit
        if (line == "-1")
            break;

        // Write line in file
        fio << line << endl;
    }

    // Execute a loop untill EOF (End of File)
    // point read pointer at beginning of file
    fio.seekg(0, ios::beg);

    while (fio) {

        // Read a Line from File
        getline(fio, line);

        // Print line in Console
        cout << line << endl;
    }

    // Close the file
    fio.close();

    return 0;
}
```


Recommended Posts:

[tellp\(\) in file handling with c++ with example](#)

[Four File Handling Hacks which every C/C++ Programmer should know](#)

[Set position with seekg\(\) in C++ language file handling](#)

[C program to copy contents of one file to another file](#)

[C++ | Signal Handling](#)

[Exception Handling in C++](#)

[Comparison of Exception Handling in C++ and Java](#)

[Exception handling and object destruction | Set 1](#)

[Nested Classes in C++](#)

[Local Classes in C++](#)

[Anonymous classes in C++](#)

[C++ Classes and Objects](#)

[Trivial classes in C++](#)

[Socket Programming in C/C++: Handling multiple clients on server without multi threading](#)

[Storage Classes in C++ with Examples](#)



[prakhargvp](#)

Check out this Author's [contributed articles](#).



If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please Improve this article if you find anything incorrect by clicking on the "Improve Article" button below.

Article Tags : [C++](#) [cpp-file-handling](#)

Practice Tags : [CPP](#)



9

3.3☐ To-do ☐ DoneBased on **8** vote(s)

Feedback/ Suggest Improvement

Add Notes

Improve Article

Please write to us at contribute@geeksforgeeks.org to report any issue with the above content.

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments

GeeksforGeeks
A computer science portal for geeks

5th Floor, A-118,
Sector-136, Noida, Uttar Pradesh - 201305
feedback@geeksforgeeks.org

COMPANY

About Us
Careers
Privacy Policy
Contact Us

LEARN

Algorithms
Data Structures
Languages
CS Subjects
Video Tutorials

PRACTICE

Courses
Company-wise
Topic-wise
How to begin?

CONTRIBUTE

Write an Article
Write Interview Experience
Internships
Videos



@geeksforgeeks, Some rights reserved