**File I/O Functions**

These are some useful classes for working with files in C++

* fstream - A combination of ofstream and ifstream: creates, reads, and writes to files
* ofstream  - creates and writes to files
* ifstream  - reads from files

The fstream library allows us to handles files. So, to be able to use files in a program, one must include the <fstream> header file.

**A. Opening a file**

In order to work with files in C++, we will first have to open it. Primarily, there are two ways to open a file:

* Using the constructor
* Using the member function open() of the class

A file could be opened for a number of uses. It could be to write to it or to read from it.

Consider an example that demonstrates the opening of a file using a constructor.

#include <iostream>

#include <fstream>

using namespace std;

int main()

{

ofstream out("example.txt");

return 0;

}

The file example.txt gets created if it’s not already there in the system and opened. Object out gets created of the type ofstream, which means it could only be used to write into the opened file.

This is how we use the constructor ofstream to open a file. Another example demonstrates the use of the ifstream constructor.

#include <iostream>

#include <fstream>

using namespace std;

int main()

{

ifstream in("example.txt");

return 0;

For this to work, a file named example.txt must already be created and present in the same folder as that of the program. Object in gets created of the type ifstream, which means it could only be used to read from the file.

**B. Closing a file**

When working with C++, closing open files is considered a good practice. A programmer often makes the mistake of not closing an open file. This becomes crucial because files do not automatically get closed after a program uses them. The closing has to be done manually.

To close a file, we have to use the close method. This is how we use them in C++.

**Syntax:**

file\_objectname.close();

Copy

**C. Writing to a file**

Writing to a file is as easy as printing any other stuff in C++. It is very similar to what we used to do when we had to print an output in the terminal. In order to write to a file, we use the insertion operator (<<). First, we create an object of the type ofstream and pass the name of the file along with its extension to the method. And then, use the extraction operator to write stuff in the file fed to the object.

Consider an example demonstrating how we write to a file.

Example:

#include <iostream>

#include <fstream>

using namespace std;

int main()

{

string str = "Welcome\_To\_My Lecture ";

ofstream out("example.txt");

out << str;

return 0;

}

Copy

Output in the example.txt file:

Welcome\_To\_My lecture

**D. Reading a file**

Reading from a file is as easy as reading any other stuff as an input in C++.  It is very similar to what we used to do when we had to read input from the terminal. In order to read from a file, we use the extraction operator (>>). First, we create an object of the type ifstream and pass the name of the file along with its extension to the method. And then, use the extraction operator to read stuff from the file fed to the object.

Consider an example demonstrating how we read from a file.

Example:

#include <iostream>

#include <fstream>

using namespace std;

int main()

{

string str;

ifstream in("example.txt");

in >> str;

cout << str;

return 0;

}