# Your Very First Program

Consider the following program:

#include <fstream> //need this if you will doing file Inp/Output. using namespace std;

int main()

{

ofstream SaveFile("cpp-home.txt"); //Open file cpp-home.txt

//Write to the file

SaveFile << "This information is written to"; SaveFile << " file cpp-home.txt! " << endl; SaveFile << "This is the seconde line. ";

SaveFile.close(); //We are done with the file. Close it. return 0;

}

* This program will create the file cpp-home.txt in the directory from where you are executing it. It will contain the following text.

This information is written to file cpp-home.txt! This is the seconde line.

* Here is what every line means:
  + #include <fstream> - You need to include this file in order to use C++’s functions for File I/O.
  + In this file, are declared several classes, including ifstream, ofstream and fstream, which are all derived from istream and ostream.
* ofstream SaveFile(“cpp-home.txt”);

1. ofstream means “output file stream”. It creates a handle for a stream to write in a file.
2. SaveFile – that’s the name of the handle. You can pick whatever you want!
3. (“cpp-home.txt”); - opens the file cpp-home.txt, which should be placed in the directory from where you execute the program. If such a file does not exists, it will be created for you, so you don’t need to worry about that!
4. SaveFile << "This information is written to"; SaveFile << " file cpp- home.txt! " << endl;SaveFile << "This is the seconde line. "; <<” looks familiar? Yes, you’ve seen it in cout <<. This (“<<”) is a predefined operator. Similar to writing to cout, but now you are writing to file, using File handle SaveFile.
5. SaveFile.close(); - As we have opened the stream, when we finish using it, we have to close it. SaveFile is an object from class ofstream, and this class

(ofstream) has a function that closes the stream. That is the close() function. So, we just write the name of the handle, dot and close(), in order to close the file stream!

**Notice:** Once you have closed the file, you can’t access it anymore, until you open it again.

That’s the simplest program, to write in a file. It’s really easy! But as you will see later in this tutorial, there are more things to learn!

# Reading A File

#include <iostream>

#include <fstream> //need this if you will doing file Inp/Output. using namespace std;

void main() //the program starts here

{

ifstream OpenFile;

OpenFile.open("cpp-home.txt"); //open file cpp-home.txt

char ch;

while(!OpenFile.eof() ) //Until it is not end of the file

{

OpenFile.get(ch);

//read a character from file to 'ch' cout << ch; //Output 'ch' to screeen

}

OpenFile.close(); //close the file.

}

* So, lets look at some important things here.
  + while(!OpenFile.eof()) – The function eof() returns a nonzero value if the end of the file has been reached. So, we make a while loop, that will loop until we reach the end of the file. So, we will get through the whole file, so that we can read it!
  + OpenFile.get(ch); - OpenFile is the object from class ifstream. This class declares a function called get(). So, we can use this function, as long as we have an object. The get() function extracts a single character from the stream and returns it. In this example, the get() function takes just one parameter- the variable name, where to put the read character. So, after calling OpenFile.get(ch) it will read one character from the stream OpenFile, and will put this character into the variable ch.

Examples:

#include <fstream>

#include <iostream> using namespace std;

void readf(ifstream &T) //pass the file stream to the function

{

//the method to read a file, that I showed you before char ch;

while(!T.eof())

{

T.get(ch); cout << ch;

}

cout << endl << "--------" << endl;

}

void main()

{

ifstream T("file1.txt"); readf(T);

T.close();

ifstream Z; Z.open("file2.txt"); readf(Z);

Z.close();

}

So, as long as file1.txt and file2.txt exists and has some text into, you will see the text being outputted.

Example: Copying one file to another.

#include <fstream>

#include <iostream>

#include <string> using namespace std;

void main()

{

ifstream T; ofstream Z; char ch; string In, Out;

cout << "Enter the name of the file to read from: " ; cin>>In;

cout << "\nEnter the name of the file to write to: " ; cin>>Out;

T.open(In.c\_str()); //open file .. Converts Strings to C-strings!!! if (!T.is\_open() ){

cout << "Can't open input file " << In << endl; cout << "Try again \n" ;

exit(1);

}

Z.open(Out.c\_str());

if (!Z.is\_open() ) {

cout << "Can't open output file " << Out << endl; cout << "Try again \n" ;

exit(1);

}

/\*while (T.get(ch)) //when it reaches to eof, it returns false and the loop stops.

Z.put(ch);

\*/

//The following is equivalent to the above statement.

T.get(ch); // attempt to get a character. If eof, stop looping. Else, write to file. while (!T.eof()){

Z.put(ch);

T.get(ch));

}

T.close();

Z.close();

cout << "Done \n";

}

**Questions:**

1. Using the notepad on your computer, create a file named "input.txt" that contains the following values

10

20

30

40

50

60

Using a while loop, read the data from the file and calculate the average of the values in the file "input.txt"

1. Append 70, 80 and 90 to the "input.txt" and test your program written for question

#4.

1. Using your notepad, create a file named **example.txt**. Type the following data into the file and save your file:

This is a sample input file.

This line is followed by 5 integer numbers: 3 2 5 90 20

* 1. Write a program that reads each line from the example.txt file and displays each line on the screen.
  2. Write a program that reads one line at a time from the example.txt file and outputs each line to the file **results.dat**.

1. The following program outputs its own C++ source file to the screen. The name of the program is called Ex1.cpp

#include <iostream>

#include <fstream> using namespace std;

int main()

{

char character; ifstream in\_stream;

in\_stream.open("Ex1.cpp"); in\_stream.get(character);

for ( ; ! in\_stream.fail() ; ) /\* alternative: while (! in\_stream.eof()) \*/

{

cout << character; in\_stream.get(character);

}

in\_stream.close();

return 0;

}

Modify the program such that it counts

* 1. the number of characters
  2. the number of words
  3. the number of lines

1. What screen output does the following program produce, and why?

#include <iostream>

#include <fstream>

using namespace std; int main()

{

char character; int integer;

ofstream out\_stream; ifstream in\_stream;

/\* Create a file containing two integers \*/ out\_stream.open("Integers");

out\_stream << 123 << ' ' << 456; out\_stream.close();

/\* Attempt to read a character, then an integer, then a character again, then an integer again, from the file "Integers" just created. \*/

in\_stream.open("Integers"); in\_stream >> character >> integer;

cout << "character: '" << character << "'\n"; cout << "integer: " << integer << "\n"; in\_stream >> character >> integer;

cout << "character: '" << character << "'\n"; cout << "integer: " << integer << "\n"; in\_stream.close();

return 0;

}