

Remember "Reviewing C++" has many more examples and topics.

Tutorial 9 **File Input / Output**

Basics

We have been dealing with input and output streams this whole time which are *cin* and *cout* respectively. The input/output stream is used with the `<iostream>` library.

In C++, as well as any programming language, you can read and write to a text file (.txt extension). This is done in a similar way to the input/output stream in terms of the library to include with the program. C++ provides 3 of them:

```
#include <ifstream>
    Library to specifically deal with input
    of text files. Does not deal with
    output.

#include <ofstream>
    Library to specifically deal with output
    of text files. Does not deal with
    input.

#include <fstream>
    Library to deal with BOTH input and output.
```

One important thing to note is that the input is treated the same as *cin* while the output is treated the same as *cout*.

Output

In order to do both input or output, you must open a file before anything else occurs. The same as a variable, the stream you are using is a type. So here is a short program that shows some simple output to a text file.

Example 1: **Simple output**

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

int main () {

    //declare the variable of type ofstream
    //since you are dealing with output:
```

Remember "Reviewing C++" has many more examples and topics.

```
ofstream myfile;

//function to open the file which includes
//the file name:
myfile.open ("example.txt");

//check if the file is open with the is_open()
//function:
if(myfile.is_open()){

    //perform the operation(s):
    myfile << "Hello world! This is output!" << endl;

    //function to close the file:
    myfile.close();
}else{

    //is_open() returned false and there is a problem:
    cout << "Can't open the file!" << endl;
}

return 0;
}
```

The above program will simply open a text file for output (ofstream), check to see if the file is correctly opened, write a line to the file and close the stream.

Here is a bit more detail about the above programs functions.

open()

The open() function will simply open a file for either input or output. The above program did it for output. The parameter of the function will be the file name. If the file does not exist in the directory, C++ will create it for you.

close()

A simple function designed to close the file and it's stream. It will require no parameters.

is_open()

The is_open() function is a boolean function that will check whether or not a file is open. If the function returns true, the file is open without any problems. If it returns false, the file is not good and therefore cannot be used.

Remember "Reviewing C++" has many more examples and topics.

Input

Recalling from the previous tutorial about the `getline()` function, input from text files are used in that fashion. Input of files also require the `<string>` library for the `getline()` function.

Here is an example of file input. The file must be created first in order for this to work.

Example 2: Simple input

```
// reading a text file
#include <iostream>
#include <fstream>
#include <string>
using namespace std;

int main () {
    string line;

    //the variable of type ifstream:
    ifstream myfile ("example.txt");

    //check to see if the file is opened:
    if (myfile.is_open())
    {
        //while there are still lines in the
        //file, keep reading:
        while (! myfile.eof() )
        {
            //place the line from myfile into the
            //line variable:
            getline (myfile,line);

            //display the line we gathered:
            cout << line << endl;
        }

        //close the stream:
        myfile.close();
    }

    else cout << "Unable to open file";

    return 0;
}
```

This program will attempt to read a file for input, display the line you read and then close the file.

Remember "Reviewing C++" has many more examples and topics.

Here is some information about the new function `eof()` which stands for "end of file".

eof()

The `eof()` function is a boolean function that will check whether or not the file has reached the end. It returns *true* when the file is at the end and *false* otherwise.