MODULE 7
FILE HANDLING

LESSON#1

The FILE class

OBJECTIVES

Upon completion of this subtopic, you will be able to:

- To understand and use different file handling functions.
- To open, create and close a file using FILE Class libraries

File Handling

Files are a means to store data in a storage device.

C++ file handling provides a mechanism to store output of a program in a file and read from a file on the disk

fopen()

The **fopen() function** in C++ opens a specified file in a certain mode.

FILE* fopen (const char* filename, const char* mode);

The **fopen()** function takes a two arguments and returns a file stream associated with that file specified by the argument filename

fopen() Parameters

filename: Pointer to the string containing the name of the file to be opened.

mode: Pointer to the string that specifies the mode in which file is opened.

File access modes

"r"	Opens the file in read mode
"w"	Opens the file in write mode
"a"	Opens the file in append mode
"r+"	Opens the file in read and write mode
"w+"	Opens the file in read and write mode
"a+"	Opens the file in read and write mode

fopen() Return value

If successful, the fopen() function returns a pointer to the FILE object that controls the opened file stream.

On failure, it returns a null pointer.

C++ fclose()

The fclose() function in C++ closes the given file stream.

int fclose(FILE* stream);

The **fclose()** function takes a single argument, a file stream which is to be closed. All the data that are buffered but not written are flushed to the OS and all unread buffered data are discarded.

Opening a file in write mode using fopen()

```
#include <cstdio>
#include <cstring>
using namespace std;
int main()
{
        int c;
        FILE *fp;
        fp = fopen("file.txt", "w");
        char str[20] = "Hello World!";
        if (fp)
        {
                for(int i=0; i<strlen(str); i++)
                        putc(str[i],fp);
        }
        fclose(fp);
}
```

Opening a file in read mode using fopen()

```
#include <cstdio>
using namespace std;
int main()
{       int c;
        FILE *fp;
        fp = fopen("file.txt", "r");
        if (fp)
        {
                while ((c = getc(fp)) != EOF)
                        putchar(c);
                fclose(fp);
        }
        return 0;
}
```

Opening a file in append mode using fopen()

```
#include <cstdio>
#include <cstring>
using namespace std;
int main()
{       int c;
        FILE *fp;
        fp = fopen("file.txt", "a");
        char str[20] = "Hello Again.";
        if (fp)
        {       putc('\n',fp);
                for(int i=0; i<strlen(str); i++)
                        putc(str[i],fp);
        }
        fclose(fp);
}
```

C++ putc()

The **putc() function** in C++ writes a character to the given output stream.

int putc(int ch, FILE* stream);

The **putc()** function takes a output file stream and an integer as its arguments. The integer is converted to unsigned char and written to the file.

Sample program

```
#include <cstdio>
#include <cstring>

int main()
{   char str[] = "Testing putc() function";
    FILE *fp;
    fp = fopen("file.txt","w");
      if (fp)
    {  for(int i=0; i<strlen(str); i++)
        {   putc(str[i],fp);       }
    }
    else
        perror("File opening failed");
        fclose(fp);
    return 0;
}
```

C++ fputc()

The fputc() function in C++ writes a character to the given output stream.

int fputc(int ch, FILE* stream);

The **fputc()** function takes a output file stream and an integer as its arguments. The integer is converted to unsigned char and written to the file.

Sample Program

```
#include <cstdio>
#include <cstring>
int main()
{  char str[] = "Hello C++ programmers";
    FILE *fp;
    fp = fopen("file.txt","w");
    if (fp)
    {  for(int i=0; i<strlen(str); i++)
        {  fputc(str[i],fp);    }
    }
    else
        perror("File opening failed");

    fclose(fp);
    return 0;
}
```

C++ fprintf()

The fprintf() function in C++ is used to write a formatted string to file stream.

int fprintf(FILE* stream, const char* format, ...);

The **fprintf()** function writes the string pointed to by format to the stream. The string format may contain format specifiers starting with % which are replaced by the values of variables that are passed to the **fprintf()** function as additional arguments.

Sample Program

```
#include <cstdio>
int main()
{  FILE *fp;
    fp = fopen("example.txt","w");
    char lang[5][20] = {"C","C++","Java","Python","PHP"};
    fprintf(fp,"Top 5 programming language\n");
    for (int i=0; i<5; i++)
        fprintf(fp, "%d. %s\n", i+1, lang[i]);
    fclose(fp);
    return 0;
}
```

REFERENCES

https://fresh2refresh.com/c-programming/c-file-handling/fopen-fclose-gets-fputs-functions-c/
https://www.programiz.com/cpp-programming/library-function/cstdio/fopen
https://www.programiz.com/cpp-programming/library-function/cstdio/putc
https://www.programiz.com/cpp-programming/library-function/cstdio/fputc
https://www.programiz.com/cpp-programming/library-function/cstdio/fprintf

LESSON#2

The fstream Library

OBJECTIVES

Upon completion of this subtopic, you will be able to:

- To open, create and close a file using the fstream library.

The fstream

The **fstream** library allows us to work with files.

To use the fstream library, include both the standard <iostream> **AND** the <fstream> header file:
Example:

```
#include <iostream>
#include <fstream>
```

The fstream

ofstream

Creates and writes to files

Ifstream

Reads from files

fstream

A combination of ofstream and ifstream: creates, reads, and writes to files

Create and Write To a File

To create a file, use either the **ofstream** or **fstream** object, and specify the name of the file.
To write to the file, use the insertion operator (<<).

```
#include <iostream>
#include <fstream>
using namespace std;
int main() {
  ofstream MyFile("filename.txt");  // Create and open a text file
  MyFile << "Hello there!";         // Write to the file
  MyFile.close();                  // Close the file
}
```

Read a File

To read from a file, use either the **ifstream** or **fstream** object, and the name of the file.

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
    string myText;
    ifstream MyReadFile("filename.txt"); // Read from the text file
    while (getline (MyReadFile, myText))
    {
      cout << myText;   // Output the text from the file
    }
    MyReadFile.close(); // Close the file
}
```

Using flags

Sr.No	Mode Flag & Description
1	**ios::app** Append mode. All output to that file to be appended to the end.
2	**ios::ate** Open a file for output and move the read/write control to the end of the file.
3	**ios::in** Open a file for reading.
4	**ios::out** Open a file for writing.
5	**ios::trunc** If the file already exists, its contents will be truncated before opening the file.

Append a File

To append a file, **fstream** is applied.

```
#include <iostream>
#include <fstream>
using namespace std;
int main ()
{
  fstream fs;
  fs.open ("filename.txt", fstream::in | fstream::out | fstream::app);
  fs << "Apppend mode! ";
  fs.close();
  return 0;
}
```

Truncate a File

If the file already exists, its contents will be truncated before opening the file.

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
  ofstream MyFile;
  MyFile.open("filename.txt", ios::out | ios::trunc);
  MyFile << "Files can be tricky, but it is fun enough!\n";
  MyFile.close();
}
```

Delete a File

To delete a file, you can use the stdio.h for removing or deleting a file.

```
#include <iostream>
#include <stdio.h>
using namespace std;
int main ()
{
  if( remove( "sample.txt" ) != 0 )
    cout <<"Error deleting file" ;
  else
    cout << "File successfully deleted" ;
  return 0;
}
```

Another Example: Writing a File

```cpp
#include <iostream>
#include <fstream>
using namespace std;
int main ()
{
  ofstream filestream("filename.txt");
  if (filestream.is_open())
  {
    filestream << "Welcome C++ Tutorial.\n";
    filestream << "Hello user.\n";
    filestream.close();
  }
  else
  {
    cout <<"No Such File created.";
  }
  return 0;
}
```

Output:

```
Welcome C++ Tutorial.
Hello user.
```

Another Example: Writing a File

```
#include <iostream>
#include <fstream>
using namespace std;
int main ()
{
  string srg;
  ifstream filestream("filename.txt");
  if (filestream.is_open())
  {
    while ( getline (filestream,srg) )
    {
      cout << srg <<endl;
    }
    filestream.close();
  }
  else
    cout << "No such file found."<<endl;
  return 0;
}
```

Output:

```
Welcome C++ Tutorial.
Hello user.
```

Writing a File based on Input

```
#include <iostream>
#include <fstream>
#include <string>
using namespace std;
int main ()
{
  string str;
  ofstream filestream("sample.txt");
  cout << "Enter a text : ";
  getline(cin,str);
  if (filestream.is_open())
  {
    filestream << str;
    filestream.close();
  }
  else
  cout <<"No Such File created.";
  return 0;
}
```

Output:

```
Enter a text : I love programming!
```

```
I love programming!
```

REFERENCES

http://www.cplusplus.com/doc/tutorial/files/
https://www.w3schools.com/cpp/cpp_files.asp
https://www.tutorialspoint.com/cplusplus/cpp_files_streams.htm
https://www.phptpoint.com/c-file-streams/
https://www.edureka.co/blog/file-handling-in-cpp/
https://www.cprogramming.com/tutorial/lesson10.html