File Handling

Course Code: CSC1102 &1103 Course Title: Introduction to Programming

Dept. of Computer Science
Faculty of Science and Technology

Lecturer No:	**15**	**Week No:**	**11(1X1.5)**	**Semester:**	
Lecturer:	*Name & email*				

File Handling

- Files are used to store data in a storage device permanently. When a program runs, the data is in the memory but when it ends or the computer shuts down, it gets lost. To keep data permanently, we need to write it in a file.

- **File handling** provides a mechanism to store the output of a program in a file and to perform various operations on it.

- Many real-life scenarios are there that handle a large number of data, and in such situations, you need to use some secondary storage to store the data. The data are stored in the secondary device using the concept of files.

Streams

- In C++ programming we are using the iostream standard library, it provides cin and cout methods for reading from input and writing to output respectively.

- To read and write from a file we are using the standard C++ library called fstream. Let us see the data types define in fstream library is:

Object/Data Type	Description
ofstream	Creates and writes to files
ifstream	Reads from files
fstream	A combination of ofstream and ifstream: creates, reads, and writes to files

Operations in File Handling

C++ provides us with the following operations in File Handling:

- Creating a file: open()
- Reading data: read()
- Writing new data: write()
- Closing a file: close()

OPENING A FILE

- A file must be opened before you can read from it or write to it. Either ofstream or fstream object may be used to open a file for writing. And ifstream object is used to open a file for reading purpose only.

- Following is the standard syntax for open() function, which is a member of fstream, ifstream, and ofstream objects.

```
open (filename, mode);
```

FILE OPENING MODES

There are some mode flags used for file opening. These are:

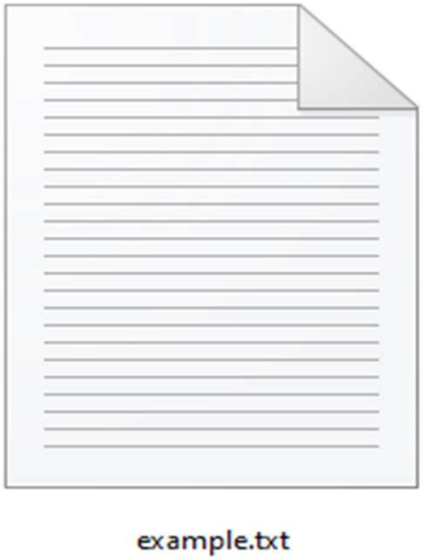
Mode	Description
ios::in	Opens the file to read(default for ifstream)
ios::out	Opens the file to write(default for ofstream)
ios::binary	Opens the file in binary mode
ios::app	Opens the file and appends all the outputs at the end
ios::trunc	Removes the data in the existing file
ios::ate	Opens the file and moves the control to the end of the file
ios::nocreate	Opens the file only if it already exists

OPENING A FILE [example]

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

int main(){
  ofstream file;
  file.open ("example.txt");
  return 0;
}
```

Output



Closing A FILE

- When a C++ program terminates it automatically flushes all the streams, release all the allocated memory and close all the opened files. But it is always a good practice that a programmer should close all the opened files before program termination.

- Following is the standard syntax for close() function, which is a member of fstream, ifstream, and ofstream objects.

  - void close();

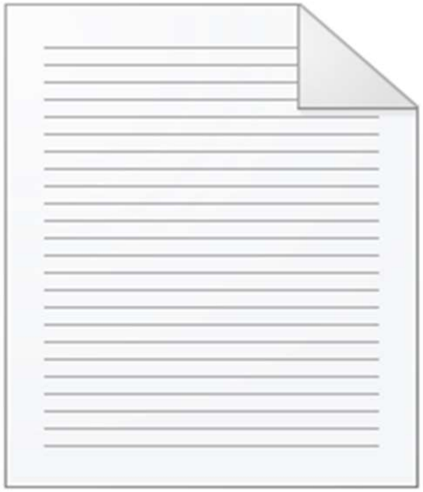
- **Why do we close the file?**
  It is considered good practice, and it can clean up unnecessary memory space.

closing A FILE [example]

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

int main(){
  ofstream file;
  file.open ("example.txt");
  file.close();
  return 0;
}
```

Output



example.txt

Write to A FILE

- While doing C++ programming, you write information to a file from your program using the stream insertion operator (<<) just as you use that operator to output information to the screen.

- The only difference is that you use an ofstream or fstream object instead of the cout object.

Write to A FILE [example]

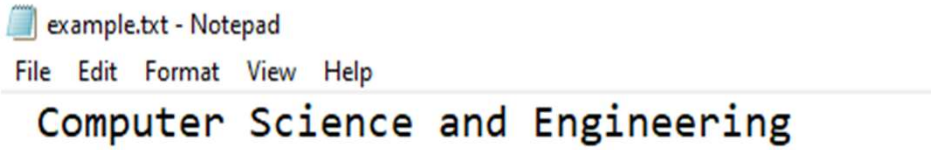
```
#include <iostream>
#include <fstream>
using namespace std;

int main(){
// Create and open a text file
  ofstream MyFile("example.txt"

// Write to the file
  MyFile << "Computer Science and Engineering";

// Close the file
  MyFile.close();
}
```

Output



```
Computer Science and Engineering
```

Read from A FILE

- While doing C++ program, programmers write information to a file from the program using the stream insertion operator (<<) and reads information using the stream extraction operator (>>).

- The only difference is that for files programmers need to use an ofstream or fstream object instead of the cout object and ifstream or fstream object instead of the cin object.

- Note that we also use a while loop together with the getline() function (which belongs to the ifstream object) to read the file line by line, and to print the content of the file.

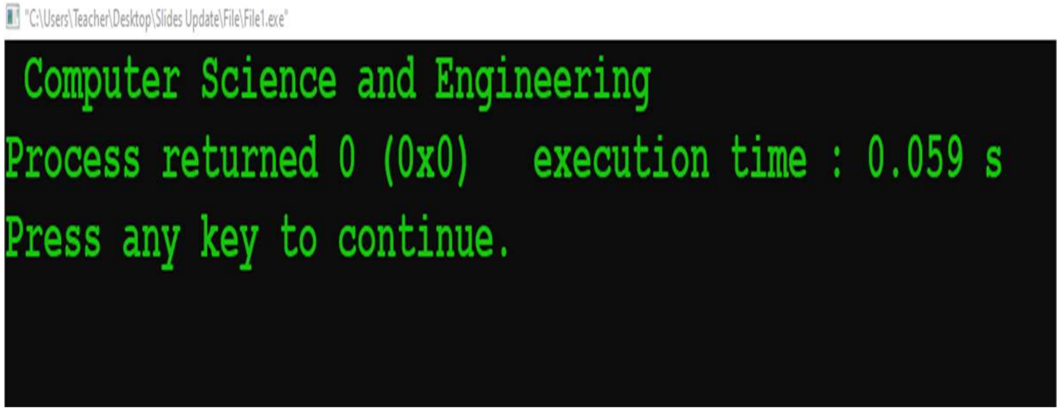
Read from A FILE [example]

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

int main(){
    string line;
    ifstream file("example.txt");

    // Read from a text file
    while(getline(file,line))
    {
        cout<<line;
    }
    file.close();
}
```

Output



```
Computer Science and Engineering
Process returned 0 (0x0)   execution time : 0.059 s
Press any key to continue.
```

Thank You