# NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES
# CL 103 - COMPUTER PROGRAMMING LAB

**Instructors:** Mr. Basit Ali, Ms.Mahrukh Khan, Ms. Ammara Yaseen, Ms. Tooba Ali, Ms. Maham Mobin, Mr. Muhammad Irfan Ayub
**Email:** basit.jasani@nu.edu.pk , mahrukh.khan@nu.edu.pk, ammara.yaseen@nu.edu.pk, tooba.ali@nu.edu.pk, maham.mobin@nu.edu.pk, muhammad.irfan@nu.edu.pk

## Lab # 10

**Outline**

- C++ Filing
- Examples
- Exercise

# C++ Filing

To perform file processing in C++, header files <iostream> and <fstream> must be included in your C++ source file.

## 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.

```
void open(const char *filename, ios::openmode mode);
```

Here, the first argument specifies the name and location of the file to be opened and the second argument of the **open()** member function defines the mode in which the file should be opened.

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

You can combine two or more of these values by **OR**ing them together. For example if you want to open a file in write mode and want to truncate it in case that already exists, following will be the syntax −

```
ofstream outfile;
outfile.open("file.dat", ios::out | ios::trunc );
```

Similar way, you can open a file for reading and writing purpose as follows −

```
fstream  afile;
afile.open("file.dat", ios::out | ios::in );
```

## 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();
```

## Writing to a File

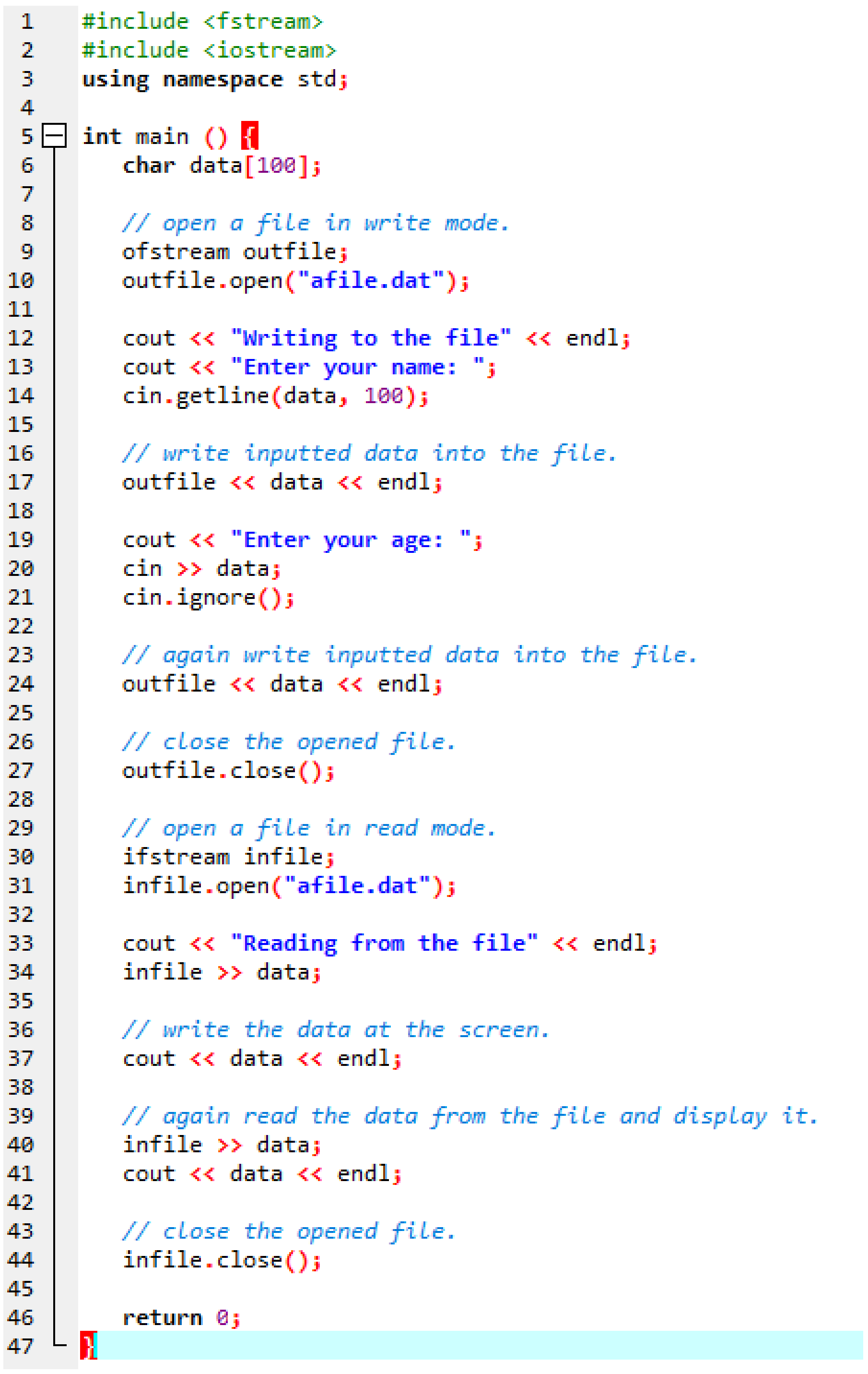
While doing C++ programming, you write information to a file from your program using the stream insertion operator (<<) just as you use that operator to output information to the screen. The only difference is that you use an **ofstream** or **fstream** object instead of the **cout** object.

## Reading from a File

You read information from a file into your program using the stream extraction operator (>>) just as you use that operator to input information from the keyboard. The only difference is that you use an **ifstream** or **fstream** object instead of the **cin** object.

### Read and Write Example

Following is the C++ program which opens a file in reading and writing mode. After writing information entered by the user to a file named afile.dat, the program reads information from the file and outputs it onto the screen.



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

int main () {
   char data[100];

   // open a file in write mode.
   ofstream outfile;
   outfile.open("afile.dat");

   cout << "Writing to the file" << endl;
   cout << "Enter your name: ";
   cin.getline(data, 100);

   // write inputted data into the file.
   outfile << data << endl;

   cout << "Enter your age: ";
   cin >> data;
   cin.ignore();

   // again write inputted data into the file.
   outfile << data << endl;

   // close the opened file.
   outfile.close();

   // open a file in read mode.
   ifstream infile;
   infile.open("afile.dat");

   cout << "Reading from the file" << endl;
   infile >> data;

   // write the data at the screen.
   cout << data << endl;

   // again read the data from the file and display it.
   infile >> data;
   cout << data << endl;

   // close the opened file.
   infile.close();

   return 0;
}
```

When the above code is compiled and executed, it produces the following sample input and output −

```
Writing to the file
Enter your name: Zara
Enter your age: 9
Reading from the file
Zara
9
```

Above examples make use of additional functions from cin object, like getline() function to read the line from outside and ignore() function to ignore the extra characters left by previous read statement.

## End of File

Returns true if the eofbit *error state flag* is set for the stream.
This flag is set by all standard input operations when the End-of-File is reached in the sequence associated with the stream.
Note that the value returned by this function depends on the last operation performed on the stream (and not on the next).
Operations that attempt to read at the *End-of-File* fail, and thus both the eofbit and the failbit end up set. This function can be used to check whether the failure is due to reaching the *End-of-File* or to some other reason.

**Return Value**
true if the stream's eofbit error state flag is set (which signals that the End-of-File has been reached by the last input operation).
false otherwise.

```
#include <iostream>     // std::cout
#include <fstream>      // std::ifstream

int main () {

  std::ifstream is("example.txt");   // open file

  char c;
  while (is.get(c))                  // loop getting single characters
    std::cout << c;

  if (is.eof())                      // check for EOF
    std::cout << "[EoF reached]\n";
  else
    std::cout << "[error reading]\n";

  is.close();                        // close file

  return 0;
}
```

# LAB 10 EXERCISES

**INSTRUCTIONS:**

**NOTE: Violation of any of the following instructions may lead to the cancellation of your submission.**

1) Create a folder and name it by your student id (k17-1234).
2) Paste the .cpp file for each question with the names such as Q1.cpp, Q2.cpp and so on into that folder.
3) Submit the zipped folder on slate.

## QUESTION#1

Write a program to implement I/O operations on characters. I/O operations includes inputting a string, calculating length of the string, Storing the String in a file and fetch the stored characters from it.

## QUESTION#2

Write a program to copy the contents of one file to another.

## QUESTION#3

Take a class Person having two attributes name and age.

- Include a parametrized constructor to give values to all data members.
- In main function
    1. Create an instance of the person class and name it person1.
    2. Create a binary file person.bin and write person1 object into it.
    3. Read the person1 object from the file.
    4. Return 0

## QUESTION#4

Take a class Participant having three attributes (ID, name and score) and following member functions

- Input () function takes data of the object and stores it in a file name participant.dat
- Output () function takes id from user and show respective data of that id.
- Max () gives the highest score of the Participant in the file.