Date :
Page :

* Object Oriented Programming (OOP) - Practical Number - 5 (Group - B)

Name :- Kaustubh Shrikant Kabra.
Class :- Second Year Engineering
Div :- A Roll Number :-
Batch :-
Department :- Computer Department
College :- AISSMS's IOIT.

Title :-

Demonstrate various file operations on file using C++.

Objectives :-

1) To learn and understand streams and files in object oriented paradigm.
2) To demonstrate file operations like create, open, read write and close a file.

Problem Statement :-

Write a C++ program that creates an output file, writes information to it, closes the file and open it again as an input file and read the information from the file.

Outcomes :-

1) Student will be able to learn and understand concepts of streams and files in C++.
2) Student will be able to demonstrate various operations like creating a new file, opening an existing file, reading from file, closing file.

Hardware requirements :-

Any CPU with Pentium processor, 256 Mb RAM or more, 1GB Hard Drive or more.

Software requirements :-

64-bit Linux/Windows Operating System, G++ compiler.

Theory :-

The iostream library is an object-oriented library that provides inputs and outputs functionality using streams.

Data Type	Description
ofstream	- This data type represents the output file stream and is used to create files and to write information to files.
Ifstream	- This data type represents the inputs file stream and is used to read information from files.
Fstream	- This data type represents the file stream generally, and has the capabilities of both ofstream and ifstream.

```
# include <iostream>
# include <fstream>
using namespace std;
int main () {
of stream myfiles;
myfile. open ("example.txt");
myfile<< "Writing this to a file. \n";
myfile. close ();
return 0;
}.
```

Opening a file:-

A file must be opened before you can read from it or write to it. Either the ofstream or fstream object may be used 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);
```

Mode Flag		Description.
ios:: app	-	Append mode. All output to that file to be appended to the end
ios:: ate	-	Open a file for output and move the read/write control to the end of the file.
ios:: in	-	Open a file for reading.
ios:: out	-	Open a file for writing.

ioos:: trunk - If the file already exists, its contents will be truncated before opening the file.

Closing a File:-

When a C++ program terminates it automatically closes flushes all the streams, release all the allocated memory and close all the opened files. Following is the standard syntax for close() function, which is a member of fstream, ifstream, and of stream objects

void close();

Writing a File:-

While doing C++ programming you write information to a file from your program using stream insertion opperator (<<) 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.

Writing operations on text files are performed in the same way we operated with cout:

```
#include < iostream>
# include< fstream>
using namespace std;
int main () {
ofstream myfile (“example. txt”);
if (myfile.is_ open())
{
myfile << “This is a line.\n”;
myfile << “This is a another line.\n”;
myfile close()
}
}
```

Reading from a file:-

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

Reading from a file can also be performed in the same way that we did with cin:

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

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

if (myfile.is_open( ) )
{
while (getline(myfile, line))
{
 cout << line <<'\n';
 }
 myfile.close ( );
}
else cout << "Unable to open file";
return 0;
}
```

Algorithm :-

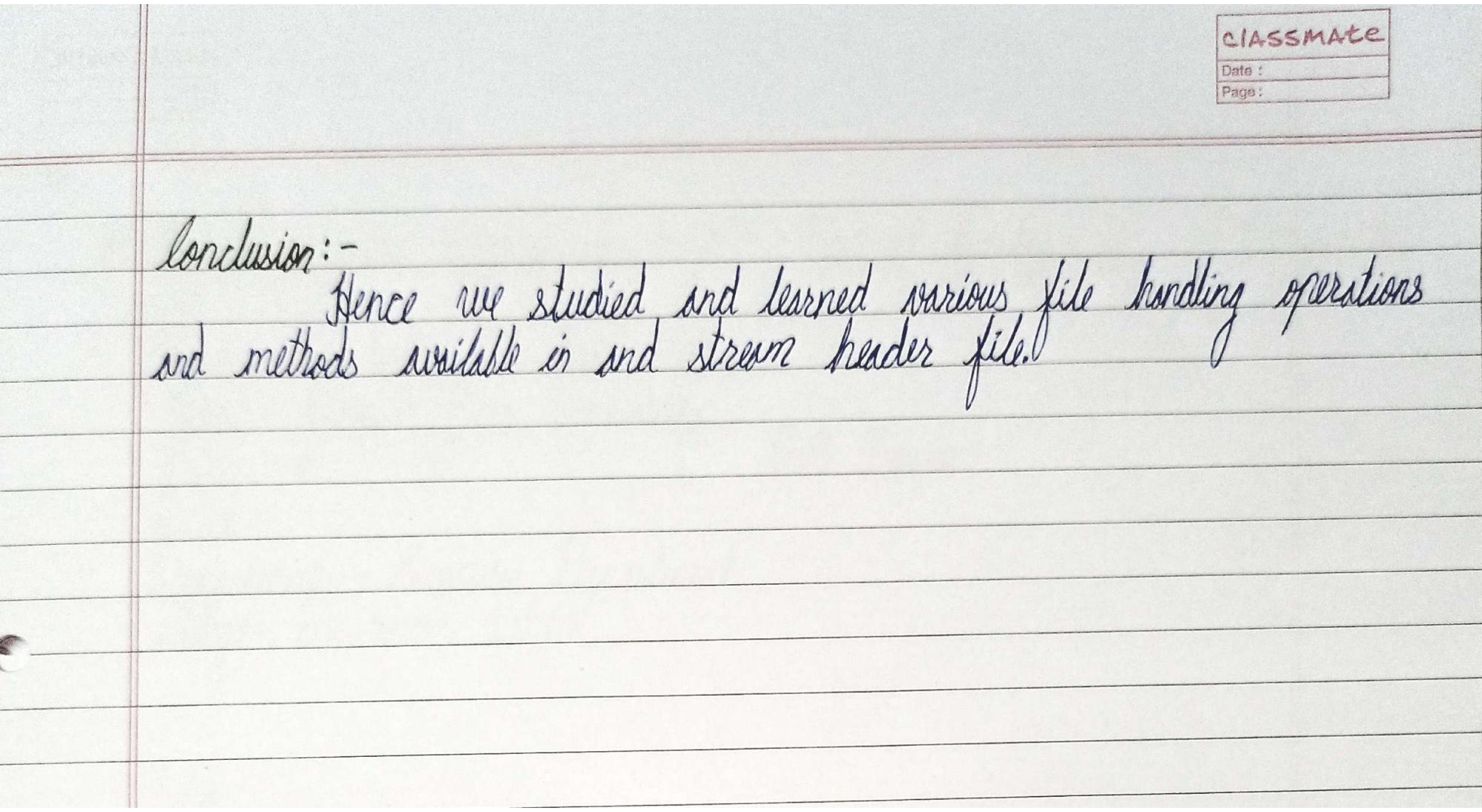
1) Start.
2) In main(), first create O/P stream and connect file using open() function.
3) Accept I/P from user
4) Write accepted data into file.
5) Close file using closes function.
6) Create I/P stream of connect file to it.
7) Check the end of the file.
8) Read data from file.
9) Display data on screen. Repeat 1 to 9.
10) If end of file detected stop reading data.
11) Close the file.
12) Stop.

Test Case :-

Enter the Data to be stored in file

Enter Student Name :- Kaustubh Kabra.
Enter Roll Number :- 20.
Enter Division :- A
Enter Class :- SE Comp-1.
Enter Contact Number :-

Data Stored in file.
Kaustubh Kabra.
20
A
SE Comp-1.



Conclusion:-

Hence we studied and learned various file handling operations and methods available in and stream header file.