EXPERIMENT NO - 12

Title: Write a program to implement File Handling

Objectives:

- **To write a data into file**
- **To read a data from file**

Key Concepts: file , streams, binary , text file

Theory:

File. The information / data stored under a specific name on a storage device, is called a file.

Stream. It refers to a sequence of bytes.

Text file. It is a file that stores information in ASCII characters. In text files, each line of text is terminated with a special character known as EOL (End of Line) character or delimiter character. When this EOL character is read or written, certain internal translations take place.

Binary file. It is a file that contains information in the same format as it is held in memory. In binary files, no delimiters are used for a line and no translations occur here.

Classes for file stream operation

ofstream: Stream class to write on files

ifstream: Stream class to read from files

fstream: Stream class to both read and write from/to files.

Opening a file

OPENING FILE USING CONSTRUCTOR

```
ofstream outFile("sample.txt");    //output only

ifstream inFile(“sample.txt”);  //input only
```

OPENING FILE USING open()

Stream-object.open("filename", mode)

```
ofstream outFile;

outFile.open("sample.txt");

ifstream inFile;

inFile.open("sample.txt");
```

File mode parameter	Meaning
ios::app	Append to end of file
ios::ate	go to end of file on opening
ios::binary	file open in binary mode
ios::in	open file for reading only
ios::out	open file for writing only
ios::nocreate	open fails if the file does not exist
ios::noreplace	open fails if the file already exist
ios::trunc	delete the contents of the file if it exist

All these flags can be combined using the bitwise operator OR (|). For example, if we want to open the file example.bin in binary mode to add data we could do it by the following call to member function open():

```
fstream file;
file.open ("example.bin", ios::out | ios::app | ios::binary);
```

Closing File

```
outFile.close();
inFile.close();
```

Basic Operation On Text File In C++ download

File I/O is a five-step process:

1. Include the header file fstream in the program.
2. Declare file stream object.
3. Open the file with the file stream object.
4. Use the file stream object with >>, <<, or other input/output functions.
5. Close the files.

Following program shows how the steps might appear in program.

Program to write in a text file

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

int main()
{
    ofstream fout;
    fout.open("out.txt");

    char str[300] = "Time is a great teacher but
            unfortunately it kills all its pupils. Berlioz";

    //Write string to the file.
    fout << str;

    fout.close();
    return 0;
}
```

Program to read from text file and display it

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

int main()
{
    ifstream fin;
    fin.open("out.txt");

    char ch;

    while(!fin.eof())
    {
        fin.get(ch);
        cout << ch;
    }

    fin.close();
    return 0;
}
```

Problem Statement :

A) 1) Write a program to write a data inside the text file

2) Write a program to read the number of characters , words, and lines from given text file.

Procedure:

- Write a program to find out number of **characters , words, and lines from given file.**
- First read the data from file using ifstream class, while reading maintain the counter to read the character , words, and lines separately
- Finally print the status for each counter
- Test the programs for two more input files.

B) Write a program to write Product_data inside the text file, and perform following operations:

- To Search Product information from file
- To Count number of products
- To Display All product information