Page No.
Date

Experiment No. 5

Name- Sumit Gulab Bhamare
Roll no.- A08
Sub- OOPL

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

Objectives: To understand the concept of file handling.
To understand the inbuilt file handling functions.

Software used:
Linux Operating Systems, GCC

Theory: So far, we have been using using the iostream standard library, which provides cin & cout methods for reading from standard input & writing to standard output respectively

Operations: Creating a file & output some data.

Data Types & Description

1. ofstream: This data type represents the output file stream & is used to create files & to write information to files.

2. ifstream: This data type represents the input file stream & is used to read information from files.

3. fstream : This data type represents the file stream generally, & has the capabilities of both ofstream & ifstream which means it can create files, write information to files, & read information from files.

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, & ofstream objects.

void open(const char *filenam, ios::openmode);

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

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 & 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.

You can combine two or more of these values by ORing them together. For example if you want to open a file in write mode & 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 & writing purpose as follows -

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

Closing a file :-
When a C++ program terminates it automatically flashes all the streams, release all the allocated memory & close all the opened files before program termination.
Following is the standard syntax for close() function, which is a member of fstream, ifstream, & 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.

Page No.
Date

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.

Algorithm:
1. Start
2. Create a file pointer object in main program.
3. With pointer object open a file in writing mode & write the content into it.
4. Close a file.
5. With pointer object open a file in reading mode & read the content from the file.
6. Close a file.
7. Stop.

Output: 1) Writing the content into file.
2) Reading the content from the file.

Conclusion: Thus, we studied concepts of file handling & its operation to perform reading the content from the file & writing the content into the file.

Program:

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


int main(){

    char Give_Info[200];
    ofstream Out_file;
    Out_file.open("Information.txt",ios::out);
    cout<<"Enter information to store it in file.(Not more than 200
characters): \n";
    fgets(Give_Info,200,stdin);
    Out_file<<Give_Info;
    Out_file.close();


    char Get_Info[200];
    ifstream In_file;
    In_file.open("Information.txt",ios::in);
    In_file.getline(Get_Info,200);
    cout<<"\nGiven information is :\n";
    cout<<Get_Info<<endl;
    In_file.close();

    Out_file.open("Information.txt",ios::app);
    cout<<"Enter information to store it in file.(Not more than 200
characters): \n";
    fgets(Give_Info,200,stdin);
    Out_file<<Give_Info;
    Out_file.close();

    In_file.open("Information.txt",ios::in);
    In_file.getline(Get_Info,200);
    cout<<"\nGiven information is :\n";
    cout<<Get_Info<<endl;
    In_file.close();


    // Out_file.open("Information.txt",ios::trunc);
    // cout<<"Enter information to store it in file.(Not more than 200
characters): \n";
    // fgets(Give_Info,200,stdin);
    // Out_file<<Give_Info;
    // Out_file.close();

    // In_file.open("Information.txt",ios::in);
```

```
    // In_file.getline(Get_Info,200);
    // cout<<"\nGiven information is :\n";
    // cout<<Get_Info<<endl;
    // In_file.close();


    return 0;
}
```

Output: