**Assignment 10- Sequential file**

AIM: Department maintains a student information. The file contains roll number, name, division and address. Allow user to add, delete information of student. Display information of particular employee. If record of student does not exist an appropriate message is displayed. If it is, then the system displays the student details. Use sequential file to main the data.

**Theory:**

File is a collection of data or a set of characters or may be text or a program. There are two types of files in C++ - Sequential and Random access files.  
  
A stream is a sequence of bytes. It is a general name given to flow of data.  
  
The stream that supplies data to the program is known as input stream. It reads the data from the file and hands over to the program.  
  
The stream that receives data from the program is known as output stream. It writes the received data to file. The classes used for input and output to the video display and keyboard are declared in the header file iostream.h. The classes used for disk file I/O are declared in the file fstream.h. The classes ofstream and ifstream are derived from mother file fstream.h. The file iostream.h is also derived from fstream.h so there is no need to include this file while handling file manipulations. fstream.h takes care of all I/O operations.  
The following classes are used in C++ to read and write files.  
**fstream: This header file is used for including the following data types.**

**ofstream: o**utput file stream is used to create files and to write information to files.

**Ifstream:** input file stream and is used to read information from files.

file stream is used for both ofstream and ifstream i.e. it can create files, write information to files, and read information from files.

|  |
| --- |
| This data type represents the output file stream and is used to create files and to write information to files. |
|  |

Ex.: ofstream outfile;

outfile.open("Student.dat",ios::app|ios::binary);

**Opening of file :**

A file must be opened before we can read from it or write to it. We can use either the **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.

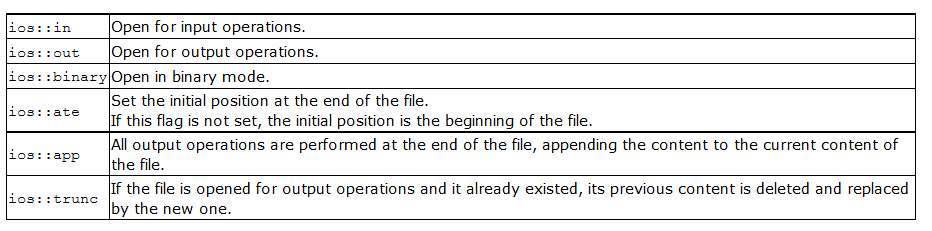
Following is the standard syntax for open() function, which is a member of fstream, ifstream, and ofstream objects.

open (filename, mode);.

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

First argument specifies the name and location of the file to be opened.

Second argument of the **open()** member function defines the mode in which the file should be opened.



File streams opened in *binary mode* perform input and output operations independently of any format considerations. Non-binary files are known as *text files.*

**Write function:** The write member function writes a given number of bytes on the given stream,

fout.write((char\*)&s1,sizeof (s1));

**read function:** The read member function extracts a given number of bytes from the given stream, placing them into the memory pointed to by the first parameter.

To read from an fstream or ifstream object, use the read method.

fin.read((char\*)&s1,sizeof (s1))

**Closing a File: is used to close**  all file stream objects

myFile.close();

## File Position Pointers

Both **istream** and **ostream** provide member functions for repositioning the file-position pointer. These member functions are **seekg** ("seek get") for istream and **seekp** ("seek put") for ostream. The argument to seekg and seekp normally is a long integer. A second argument can be specified to indicate the seek direction. The seek direction can be **ios::beg** (the default) for positioning relative to the beginning of a stream, **ios::cur** for positioning relative to the current position in a stream or **ios::end** for positioning relative to the end of a stream.

// position to the nth byte of fileObject (assumes ios::beg)

fileObject.seekg( n );

// position n bytes forward in fileObject

fileObject.seekg( n, ios::cur )

**tellg**() get the position of the get pointer and **tellp**() gets the position of the put pointer

**Conclusion:**

Performed operations on sequential file and understood the operations to be performed on sequential text file.