ASSINGMENT NO.	**11**
TITLE	**Sequential File**
PROBLEM STATEMENT /DEFINITION	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 student. 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 maintain the data.
OBJECTIVE	To understand practical implementation and usage of Sequential File for solving the problems.
OUTCOME	After successful completion of this assignment, students will be able to implement and use **Sequential File** for efficient solution to problems.
S/W PACKAGES AND HARDWARE APPARATUS USED	• (64-bit)64-BIT Fedora 17 or latest 64-bitupdate of equivalent Open source OS • Programming Tools (64-bit) Latest Open source update of Eclipse Programming frame work
REFERENCES	1. E. Horowitz S. Sahani, D. Mehata, "Fundamentals of data structures in C++", Galgotia Book Source, New Delhi, 1995, ISBN: 1678298 2. SartajSahani, ―Data Structures, Algorithms andApplications in C++‖, Second Edition, University Press, ISBN:81-7371522 X.
INSTRUCTIONS FOR WRITING JOURNAL	1. Date 2. Assignment no. and title 3. Problem definition 4. Learning Objective 5. Learning Outcome 6. Software / Hardware requirement 7. Concepts related Theory 8. Algorithms/ Pseudo Code 9. Class ADT 10. Test cases 11. Conclusion/Analysis

Prerequisites:

- Basic knowledge of File Data Structure.
- Its operations – Create, Open, Write and Read etc.

Learning Objectives:

- To understand practical implementation and usage of Sequential Access File for solving the problems.

Learning Outcomes:

- After successful completion of this assignment, students will be able to implement and use **Sequential Access File** for efficient solution to problems.

Concepts related Theory:

File is a collection of records.

File Organization: File organization ensures that records are available for processing. It is used to determine an efficient file organization for each base relation.

- **3 Types of File Organization**

1. Sequential access file organization
2. Direct access file organization
3. Indexed sequential access files organization

A Sequential file contains records organized by the order in which they were entered. The order of the records is fixed. Records in sequential files can be read or written only sequentially.
After you place a record into a sequential file, you cannot shorten, lengthen, or delete the record. However, you can update (REWRITE) a record if the length does not change. New records are added at the end of the file.

If the order in which you keep records in a file is not important, sequential organization is a good choice whether there are many records or only a few. Sequential output is also useful for printing reports.

Advantages:

- It is simple to program and easy to design.
- Sequential file is best use if storage space.

Disadvantages:

- Sequential file is time consuming process.
- It has high data redundancy.
- Random searching is not possible.

Representation:

Following is a pictorial representation of Sequential File –

Basic Operations:

Following are the basic operations of Sequential File −

- **Creating a file** − Creation of a file is also defined as the loading of the file.
- **Opening a file** − Before Read/Write, that file must be opened.

  (Name of a file and the access mode (read or write).
- **Writing a file** − A write operation to write Records into the file.
- **Reading a file** – To read Records from file sequentially.

Record:- is a collection of fields related to particular entity, such as Student, Employee etc.

File functions to be use:

```
//…..Header Files
#include<iostream>
#include<fstream>

//....File Object/Handle for Reading & Writing
fstream Myfile;

//....Character array for Getline() function
char buffer[40];
```

a) To Open Sequential File for Writing and then Close the File.

```
//Step 01: Opening File for Writing.....";
Myfile.open("File1.txt" , ios::out);

if(!Myfile)
        cout<<"\n\n There is an error Opening the file....";
else
{
                        //Step 02: The file is Opened Successfully...!!!";

                        // Step 03: Writing into the File.....\n";

                        // Roll No-Name-Division-Address

        Myfile<<"11-Nitya SE-I Computer Kothrud, Pune."<<endl;

        Myfile<<"12-Sanju SE-I Computer Ambegaon, Pune."<<endl;

        Myfile<<"13-Mitali SE-I Computer Dhayari, Pune."<<endl;
}
Myfile.close();
                //Step 04: The file has been closed successfully...!!!";
```

b) To Open Sequential File for Reading and then Close the File.

```
        //Step 05: Opening File for Reading.....";

        Myfile.open("File1.txt" , ios::in);

        if(!Myfile)
                cout<<"\n\n There is an error Opening the file....";
        else
        {
                        //Step 06: The file is Opened Successfully...!!!";

                        //Step 07: Reading from the File.....\n";

                Myfile.getline(buffer, 40);
                cout<<"\n Record-01: "<<buffer;
```

```
        Myfile.getline(buffer, 40);
        cout<<"\n Record-02: "<<buffer;

        Myfile.getline(buffer, 40);
        cout<<"\n Record-03: "<<buffer;
}
Myfile.close();
                //Step 08: The file has been closed successfully...!!!";
```

c) Deleting a FILE:

In Sequential File we cannot delete the record but can delete complete file using remove() Member function. This function need one argument which is the name of the file: the basic syntax for deleting file is:

```
remove("Myfile");
```

Algorithms and Pseudo code:

A] write_Records():

Step1: START
Step2: Create and open the Sequential file to add records.
Step3: Accept student's roll number, name, division and address.
Step4: Write the record in the file.
Step5: Repeat steps 3 – 4 for no. of records to be inserted for N students.
Step6: Close the file.
Step7: STOP

B] read_Records():

Step1: START
Step2: Create and open the Sequential file to read records.
Step3: Read student's roll number, name, division and address.
Step4: Display the record on the screen.
Step5: Repeat steps 3 – 4 for N records to be displayed.
Step6: Close the file.
Step7: STOP

Conclusion:

In this way the Sequential file is implemented to read and write records sequentially using the object of fstream class to obtain desired results.

Sample Questions:

1. Define File? Explain the file organization in C++?
2. Compare Text and Binary File?
3. Explain the different File opening modes in C++?
4. Which are the different types of Files? Explain in short.
5. Differentiate between ……
    a) Sequential Access File
    b) Direct Access File
    c) Indexed Sequential file.
6. What is a sequential access storage device?
7. How to add record in sequential file?
8. How to delete record from sequential file?
9. What are the advantages and disadvantages of Sequential file organization?
10. Distinguish between logical and physical deletion of records and illustrate it with suitable examples.