**ADVANCED DATA STRUCTURE**

**GROUP E**

**ASSIGNMENT 12**

**YEAR: 2017-18**

**COLLEGE: VIIT**

**DATE OF COMPLETION : 30/03/2018**

**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.

**Objective:**

To make use of sequential ﬁles to maintain and operation on data.

**Software And Hardware Requirement:**

1. 64-bit Open source Linux or its derivative.

2. Open Source C++ Programming tool like G++/GCC.

**Theory:**

Files are a means to store data in a storage device. C++ file handling provides a mechanism to store output of a program in a file and read from a file on the disk. So far, we have been using **<iostream>** header file which provide functions **cin** and **cout**to take input from console and write output to a console respectively. Now, we introduce one more header file **<fstream>**which provides data types or classes ( **ifstream** , **ofstream** , **fstream** ) to read from a file and write to a file.

**File Opening Modes**  
A file can be opened in different modes to perform read and write operations. Function to open a file i.e **open( )** takes two arguments : **char \*filename** and **ios :: mode**. C++ supports the following file open modes :

|  |  |
| --- | --- |
| **Mode** | **Explanation** |
| ios :: in | Open a file for reading |
| ios :: out | Open a file for writing |
| ios :: app | Appends data to the end of the file |
| ios :: ate | File pointer moves to the end of the file but allows to writes data in any location in the file |
| ios :: binary | Binary File |
| ios :: trunc | Deletes the contents of the file before opening |

If a file is opened in **ios :: out** mode, then by default it is opened in **ios :: trunc** mode also i.e the contents of the opened file is overwritten. If we open a file using **ifstream** class, then by default it is opened in **ios :: in** mode and if we open a file using **ofstream** class, then by default it is opened in **ios :: out** mode. The **fstream** class doesn’t provide any default mode.

**Algorithm:**

1.Enter a choice ’ch’ from the choice menu.

2. if ch=0, add record of a student in the file ’stud.dat’.

3. if ch=1, search record of a specific student from the file ’Data.dat’.

4.if ch=2, delete record of a specific student from the file ’Data.dat’.

5.if ch=3, then exit.

**Code :**

#include <iostream>

#include <fstream>

#include <cstring>

#include <iomanip>

#include<cstdlib>

#define max 50

using namespace std;

class Student

{

char name[max];

int rollNo;

int year;

int division;

char address[50];

friend class FileOperations;

public: Student()

{

strcpy(name,"");

rollNo=year=division=0;

strcpy(address,"");

}

Student(char name[max],int rollNo,int year,int division,char address[max])

{

strcpy(this->address,address);

strcpy(this->name,name);

this->division=division;

this->rollNo=rollNo;

this->year=year;

}

int getRollNo()

{

return rollNo;

}

void displayStudentData()

{

cout<<endl<<setw(3)<<rollNo<<setw(10)<<name<<setw(3)<<year<<setw(2)<<division<<setw(10)<<address;

}

};

class FileOperations

{

fstream file;

public:FileOperations(char \*name)

{

//strcpy(this->name,name);

this->file.open(name,ios::in|ios::out|ios::ate|ios::binary);

}

void insertRecord(int rollNo,char name[max],int year,int division,char address[max])

{

Student s=Student(name,rollNo,year,division,address);

file.seekp(0,ios::end);

file.write((char\*)&s,sizeof(Student));

file.clear();

}

void displayAllRecords()

{

Student s;

file.seekg(0,ios::beg);

while(file.read((char \*)&s,sizeof(Student)))

{

s.displayStudentData();

}

file.clear();

}

void displayRecord(int rollNo)

{

Student s;

file.seekg(0,ios::beg);

void \*p;

while(file.read((char \*)&s,sizeof(Student)))

{

if(s.rollNo==rollNo)

{

s.displayStudentData();

break;

}

}

if(p==NULL)

throw "Element not present";

file.clear();

}

void deleteRecord(int rollNo)

{

ofstream newFile("new.txt",ios::binary);

file.seekg(0,ios::beg);

bool flag=false;

Student s;

while(file.read((char \*)&s,sizeof(s)))

{

if(s.rollNo==rollNo)

{

flag=true;

continue;

}

newFile.write((char \*)&s,sizeof(s));

}

if(!flag)

{

cout<<"Element Not Present";

}

file.close();

newFile.close();

remove("student.txt");

rename("new.txt","student.txt");

file.open("student.txt",ios::in|ios::ate|ios::out|ios::binary);

}

~FileOperations()

{

file.close();

cout<<"Closing file..";

}

};

int main()

{

ofstream newFile("student.txt",ios::app|ios::binary);

newFile.close();

FileOperations file((char \*)"student.txt");

int rollNo,year,division,choice=0;

char name[max],address[max];

while(choice!=5)

{

//clrscr();

cout<<"\n\*\*\*\*\*Phone Book\*\*\*\*\*\n";

cout<<"1) Add New Record\n";

cout<<"2) Display All Records\n";

cout<<"3) Display by RollNo\n";

cout<<"4) Deleting a Record\n";

cout<<"5) Exit\n";

cout<<"Choose your choice : ";

cin>>choice;

switch(choice)

{

case 1 : //New Record

cout<<endl<<"Enter RollNo and name : \n";

cin>>rollNo>>name;

cout<<"Enter Year and Division : \n";

cin>>year>>division;

cout<<"Enter address : \n";

cin>>address;

file.insertRecord(rollNo,name,year,division,address);

break;

case 2 :

file.displayAllRecords();

break;

case 3 :

cout<<"Enter Roll Number";

cin>>rollNo;

try

{

file.displayRecord(rollNo);

}

catch(const char \*str)

{

cout<<str;

}

break;

case 4:

cout<<"Enter rollNo";

cin>>rollNo;

file.deleteRecord(rollNo);

break;

case 5 :break;

}

}

}

**Output:**

\*\*\*\*\*Phone Book\*\*\*\*\*

1) Add New Record

2) Display All Records

3) Display by RollNo

4) Deleting a Record

5) Exit

Choose your choice : 1

Enter RollNo and name :

1

shree

Enter Year and Division :

2016

2

Enter address :

sdf

\*\*\*\*\*Phone Book\*\*\*\*\*

1) Add New Record

2) Display All Records

3) Display by RollNo

4) Deleting a Record

5) Exit

Choose your choice : 1

Enter RollNo and name :

2

vijay

Enter Year and Division :

2016

1

Enter address :

wsd

\*\*\*\*\*Phone Book\*\*\*\*\*

1) Add New Record

2) Display All Records

3) Display by RollNo

4) Deleting a Record

5) Exit

Choose your choice : 2

1 shree2016 2 sdf

2 vijay2016 1 wsd

\*\*\*\*\*Phone Book\*\*\*\*\*

1) Add New Record

2) Display All Records

3) Display by RollNo

4) Deleting a Record

5) Exit

Choose your choice : 3

Enter Roll Number1

1 shree2016 2 sdf

\*\*\*\*\*Phone Book\*\*\*\*\*

1) Add New Record

2) Display All Records

3) Display by RollNo

4) Deleting a Record

5) Exit

Choose your choice : 4

Enter rollNo2

\*\*\*\*\*Phone Book\*\*\*\*\*

1) Add New Record

2) Display All Records

3) Display by RollNo

4) Deleting a Record

5) Exit

Choose your choice : 2

1 shree2016 2 sdf

\*\*\*\*\*Phone Book\*\*\*\*\*

1) Add New Record

2) Display All Records

3) Display by RollNo

4) Deleting a Record

5) Exit

Choose your choice : 5

Closing file..

**Conclusion:**

Various operations on creation, addition, search, deletion, update contents on file performed.