**Experiment No: 5 Date:- 7-11-2020**

**AIM: To study File Processing.**

**THEORY:**

Files are used to store data in a storage device permanently. File handling provides a mechanism to store the output of a program in a file and to perform various operations on it.

A stream is an abstraction that represents a device on which operations of input and output are performed. A stream can be represented as a source or destination of characters of indefinite length depending on its usage.

In C++ we have a set of file handling methods. These include ifstream, ofstream, and fstream. These classes are derived from fstreambase and from the corresponding iostream class. These classes, designed to manage the disk files, are declared in fstream and therefore we must include fstream and therefore we must include this file in any program that uses files.

In C++, files are mainly dealt by using three classes fstream, ifstream, ofstream available in fstream headerfile.  
**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.

Now the first step to open the particular file for read or write operation. We can open file by  
1. passing file name in constructor at the time of object creation  
2. using the open method

The first argument of the open function defines the name and format of the file with the address of the file.

The second argument represents the mode in which the file has to be opened. The following modes are used as per the requirements.

Operations in File Handling:

* Creating a file: open()
* Reading data: read()
* Writing new data: write()
* Closing a file: close()

**Special Operations in a file:**

There are few important functions to be used with file streams like:

* tellp() - It tells the current position of the put pointer.

**Syntax:** filepointer.tellp()

* tellg() - It tells the current position of the get pointer.

**Syntax:** filepointer.tellg()

* seekp() - It moves the put pointer to mentioned location.

**Syntax:** filepointer.seekp(no of bytes,reference mode)

* seekg() - It moves get pointer(input) to a specified location.

**Syntax:** filepointer.seekg((no of bytes,reference point)

* put() - It writes a single character to file.
* get() - It reads a single character from file.

**Note:**For seekp and seekg three reference points are passed:  
***ios::beg*** *- beginning of the file****ios::cur*** *- current position in the file****ios::end*** *- end of the file*

**A] Write a C++ program to insert 5 elements in first file and 3 elements in second file. Merge the contents of both files into third file and put it in ascending order.**

**#include<iostream>**

**#include<fstream>**

**using namespace std;**

**int main()**

**{**

**fstream a,b,c;**

**a.open("abc.txt", ios::out);**

**b.open("xyz.txt", ios::out);**

**int num;**

**cout<<"Enter 5 elements in the first file"<<endl;**

**int i=1;**

**while(i<=5)**

**{**

**cout<<"Enter element "<<i<<": ";**

**cin>>num;**

**a.write((char\*)&num,sizeof(num));**

**i++;**

**}**

**cout<<endl<<endl;**

**cout<<"Enter 3 elements in the second file"<<endl;**

**i=1;**

**while(i<=3)**

**{**

**cout<<"Enter element "<<i<<": ";**

**cin>>num;**

**b.write((char\*)&num,sizeof(num));**

**i++;**

**}**

**a.close();**

**b.close();**

**int num1, num2;**

**a.open("abc.txt", ios::in);**

**b.open("xyz.txt", ios::in);**

**c.open("sort.txt", ios::out);**

**a.seekg(0,ios::beg);**

**b.seekg(0,ios::beg);**

**while(a.read((char\*)&num1,sizeof(num1)))**

**c.write((char\*)&num1,sizeof(num1));**

**while( b.read((char\*)&num2,sizeof(num2)) )**

**c.write((char\*)&num2,sizeof(num2));**

**c.close();**

**int arr[8];**

**i=0;**

**c.open("sort.txt",ios::in);**

**while(c)**

**{**

**c.read((char\*)&num,sizeof(num));**

**arr[i]=num;**

**i++;**

**}**

**c.close();**

**cout<<"Before sorting:"<<endl;**

**for(i=0;i<8;i++)**

**cout<<arr[i]<<" ";**

**cout<<endl;**

**cout<<endl<<"After sorting:"<<endl;**

**for(i=1;i<8;i++)**

**for(int j=0;j<8-i;j++)**

**{**

**if(arr[j]>arr[j+1])**

**{**

**num=arr[j];**

**arr[j]=arr[j+1];**

**arr[j+1]=num;**

**}**

**}**

**for(i=0;i<8;i++)**

**cout<<arr[i]<<" ";**

**cout<<endl;**

**c.open("sort.txt",ios::out);**

**i=0;**

**while(i<8)**

**{**

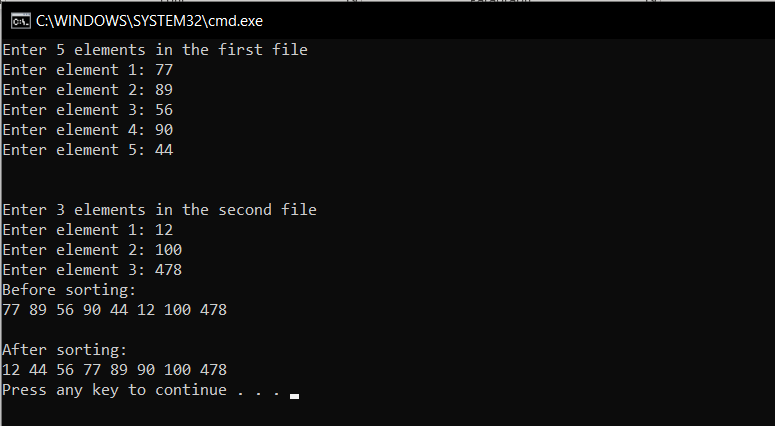
**c.write((char\*)&arr[i],sizeof(arr[i]));**

**i++;**

**}**

**a.close();**

**b.close();**

**c.close();}**

**B] Write a C++ program to simulate a telephone directory application. Program should prompt user to enter name and telephone number of users. Also the program should allow the user to search and update the telephone number of a specific user depending upon the name entered.**

#include<iostream>

#include<fstream>

#include<ctype.h>

using namespace std;

class directory

{

char name[30];

long long int n;

public: void update()

{

cin>>n;

}

void getdata()

{

cout<<"Enter name"<<endl;

cin>>name;

cout<<"Enter corresponding phone no"<<endl;

cin>>n;

}

char \* getinfo()

{

return name;

}

};

fstream tele;

void add();

void update();

int main()

{

int c;

do

{

cout<<"Enter your choice"<<endl;

cout<<"1.Add a telephone record"<<endl<<"2.Update an existing record"<<endl<<"0.exit"<<endl;

cin>>c;

switch(c)

{

case 1:add();break;

case 2:update();break;

default:if(c)

cout<<"Wrong choice"<<endl;

}

}while(c!=0);

return 0;

}

void add()

{

tele.open("directory",ios::app | ios::binary);

directory z;

z.getdata();

tele.write((char\*)&z,sizeof(z));

while(getchar()!='\n');

tele.close();

cout<<"The record was added successfully"<<endl<<endl;

}

void update()

{

char nm[30];

directory z;

tele.open("directory",ios::in | ios::binary);

tele.seekg(0);

cout<<"Enter the name of the user whose phone number is to be updated"<<endl;

cin>>nm;

int g,p=0;

while(tele.read((char\*)&z,sizeof(z)))

{

if(strcmp(nm,z.getinfo()) == 0)

{

p=1;

g=tele.tellg();

break;

}

}

tele.close();

if(p)

{

tele.open("directory", ios::ate | ios::binary);

g = g-sizeof(z);

tele.seekp(g);

cout<<"Enter the new phone number"<<endl;

z.update();

tele.write((char\*)&z,sizeof(z));

cout<<"The record was updated successfully"<<endl;

tele.close();

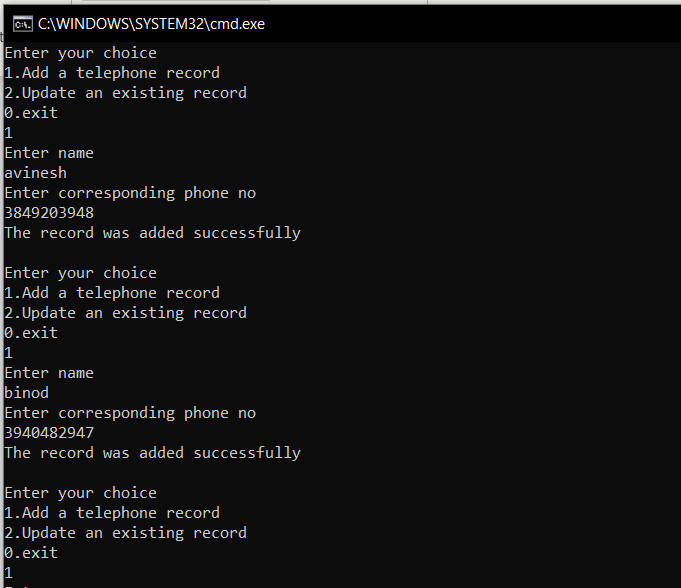
}

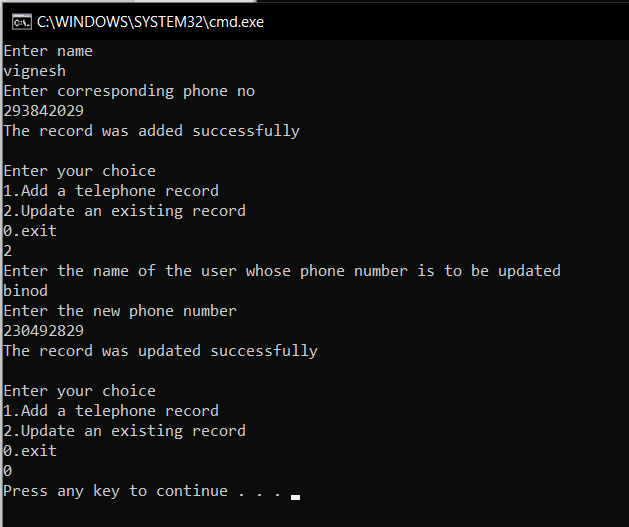
else

cout<<"The record doesnt exist"<<endl;

cout<<endl;

}

****

****

**C] Write a C++ program to create a student’s database application using “files”. Create a unique file for each student depending upon the student name entered. Store the student data like name, roll no, address, and branch into the file. Allow the user to search and update all the student details depending upon the entered roll-no and display the details.**

#include<iostream>

#include<fstream>

#include<ctype.h>

#include<iomanip>

using namespace std;

class student

{

char name[30];

int roll;

char address[45];

char branch[20];

public: void update();

void putdata();

void getdata();

char \* getname()

{

return name;

}

int getroll()

{

return roll;

}

};

void student::getdata()

{

cout<<"Enter name"<<endl;

cin>>name;

cout<<"Enter roll no"<<endl;

cin>>roll;

cout<<"Enter branch"<<endl;

cin>>branch;

while((getchar())!='\n');

cout<<"Enter address"<<endl;

cin.getline(address, 45);

//while((gectchar())!='\n');

}

void student::putdata()

{

cout<<name<<setw(10)<<roll<<setw(16)<<branch<<setw(17)<<address<<endl;

}

int main()

{

student a[20];

char c;

int i=-1;;

do

{

i++;

a[i].getdata();

fstream f(a[i].getname(),ios::out | ios::binary);

f.write((char\*)&a[i],sizeof(a[i]));

f.close();

cout<<"Do you want to save another student's record?(y/n)"<<endl;

cin>>c;

}while(c!='n');

cout<<"The records are as follows"<<endl;

cout<<"Name"<<setw(12)<<"Roll"<<setw(15)<<"Branch"<<setw(15)<<"Address"<<endl;

for(int j=0;j<=i;j++)

a[j].putdata();

int roll;

cout<<"Enter roll no. of student whose record is to be updated"<<endl;

cin>>roll;

int j=0;

while(j<=i)

{

if(roll == a[j].getroll())

break;

j++;

}

if(j>i)

{

cout<<"The record doesnt exit"<<endl;

exit(1);

}

fstream f(a[j].getname(), ios::in | ios::binary);

f.read((char\*)&a[j],sizeof(a[j]));

f.close();

remove(a[j].getname());

cout<<"Enter the new details of the student"<<endl;

a[j].getdata();

f.open(a[j].getname(), ios::out | ios::binary);

f.write((char\*)&a[j],sizeof(a[j]));

f.close();

cout<<"The record was successfully updated as shown"<<endl;

cout<<"Name"<<setw(12)<<"Roll"<<setw(14)<<"Branch"<<setw(15)<<"Address"<<endl;

for(int j=0;j<=i;j++)

{

f.open(a[j].getname(), ios::in | ios::binary);

f.read((char\*)&a[j],sizeof(a[j]));

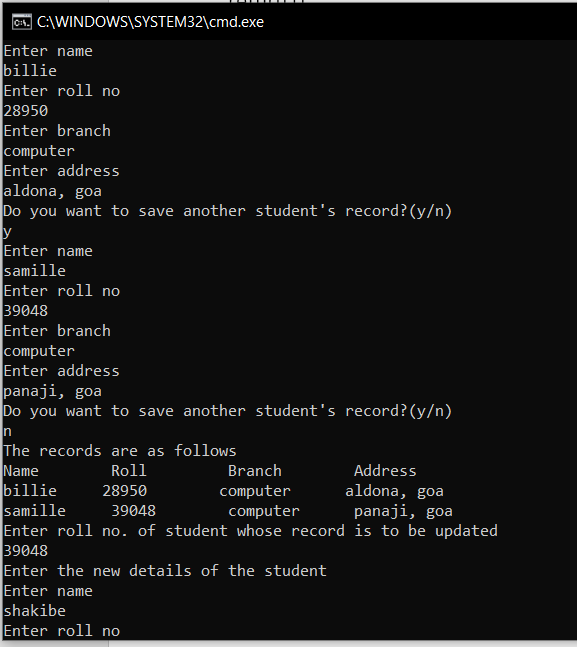
a[j].putdata();

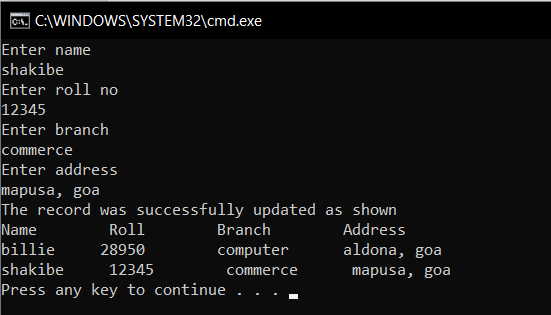
}

f.close();

return 0;

}





**CONCLUSION:**

All programs were implemented and run successfully, with sole emphasis on file handling processes.

* The open() function can be used to open multiple files that use the same stream object.
* If we do not specify the second argument of open() function, the default values specified in prototype of these class member functions are used while opening the file.
* To open an existing file for updating without losing its original contents we need to open it in an append mode.
* The fstream class does not provide a mode by default and therefore we must provide the mode explicitly when using an object of fstream class.
* We can specify more than one file mode using bitwise operators.