* **Student.h**

/\*

\* Student.h

\* Author: Megha Sonavane

\*/

**#ifndef** STUDENT\_H\_

**#define** STUDENT\_H\_

**class** Student {

**int** rollNo;

std::string name;

std::string address;

std::string div;

**public**:

**Student**();

**void** **input**();

**void** **show**();

**int** **getRollNo**();

**void** **setName**(std::string);

**void** **setAddress**(std::string);

**void** **setDivision**(std::string);

**virtual** **~Student**();

};

**#endif** /\* STUDENT\_H\_ \*/

* **Student.cpp**

/\*

\* Student.cpp

\* Author: Megha Sonavane

\*/

**#include**<iostream>

**#include** "Student.h"

**using** **namespace** std;

**Student::Student**() {

// **TODO** Auto-generated constructor stub

}

//===================definition of input()===========================

**void** **Student::input**(){

cout<<"Enter details of student:"<<**endl**;

cout<<"Roll No:";

cin>>rollNo;

cin.ignore();

cout<<"Name:";

**getline**(cin,name);

cout<<"Division:";

**getline**(cin,div);

cout<<"Address:";

**getline**(cin,address);

}

//=================definition of show()===============================

**void** **Student::show**(){

cout<<rollNo<<"\t"<<name<<"\t"<<div<<"\t"<<address<<**endl**;

}

//=========definition of getRollNo()=======================================

**int** **Student::getRollNo**(){

**return** **this**->rollNo;

}

//=====definition of setName()=========================================

**void** **Student::setName**(string nm)

{

**this**->name=nm;

}

//=====definition of setAddress()===========================================

**void** **Student::setAddress**(string add){

**this**->address=add;

}

//====definition of setDivision()==============================================

**void** **Student::setDivision**(string d){

**this**->div=d;

}

**Student::~Student**() {

// **TODO** Auto-generated destructor stub

}

* **FileOperation.h**

/\*

\* FileOperation.h

\* Author: Megha Sonavane

\*/

**#include** "Student.h"

**#ifndef** FILEOPERATION\_H\_

**#define** FILEOPERATION\_H\_

**using** **namespace** std;

**class** FileOperation {

Student student;

**public**:

**FileOperation**();

**void** **create**();

**void** **addnew**();

**void** **display**();

**void** **search**();

**void** **modify**();

**void** **delet**();

**virtual** **~FileOperation**();

};

**#endif** /\* FILEOPERATION\_H\_ \*/

* **FileOperation.cpp**

/\*

\* FileOperation.cpp

\* Author: Megha Sonavane

\*/

**#include**<iostream>

**#include**<fstream>

**#include** "FileOperation.h"

**FileOperation::FileOperation**() {

// **TODO** Auto-generated constructor stub

}

//===============definition of create()======================================

**void** **FileOperation::create**(){

//Open file in write mode to crete new database

fstream file("StudDB.dat", ios::*out* | ios::*binary*);

cout<<"New database created..."<<**endl**;

//add record to it

addnew();

file.close();

}

//===============Definition of AddNew()======================================

**void** **FileOperation::addnew**(){

//read data to be added in student object

student.input();

//open file in append mode

fstream file("StudDB.dat",ios::*app* | ios::*binary*);

//----validating roll number---------

**int** flag=1; //to validate roll number

//open file in read mode with different pointer

fstream file1("StudDB.dat",ios::*in* | ios::*binary*);

Student s1; //to read records from file

file1.read((**char**\*)&s1,**sizeof**(s1));

//read file till end

**while**(!file1.eof())

{

//check if roll number is present

**if**(s1.getRollNo()==student.getRollNo())

{

cout<<"Student with roll number"<<student.getRollNo()<<" already exist..."<<**endl**;

flag=0;

**break**;

}

//else read next record

**else**

file1.read((**char**\*)&s1,**sizeof**(s1));

}

//if roll number is not exist already

**if**(flag==1){

//write to file

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

cout<<"Added successfully.."<<**endl**;

}

file.close();

}

//============definition of display()==========================================

**void** **FileOperation::display**(){

//open file in read mode

fstream file;

file.open("StudDB.dat",ios::*in* | ios::*binary*);

//if file not found

**if**(!file)

{

cout<<"File not found"<<**endl**;

**exit**(1);

}

**else**

{

//read first record

file.read((**char**\*)&student,**sizeof**(student));

cout<<"RollNo\tName\tDivision\tAddress"<<**endl**;

//display records till end of file

**while**(!file.eof())

{

student.show(); //display cuurent record

file.read((**char**\*)&student,**sizeof**(student)); //read next record

}

}

file.close();

}

//===========definition of search()==================================================

**void** **FileOperation::search**(){

//open file in read mode

fstream file("StudDB.dat", ios::*in* | ios::*binary*);

//if file not found

**if**(!file)

{

cout<<"File not found..";

**exit**(1);

}

**else**{

**int** key;

**bool** flag=**false**;

cout<<"Enter roll number to be search:";

cin>>key;

//read first record

file.read((**char**\*)&student,**sizeof**(student));

//search till end of file

**while**(!file.eof()){

//if key is equal to current roll number

**if**(student.getRollNo()==key)

{

cout<<"Record found.."<<**endl**;

flag=**true**;//set flag

cout<<"RollNo\tName\tDivision\tAddress"<<**endl**;

student.show(); //display record

**break**;

}

**else** //read next record

file.read((**char**\*)&student,**sizeof**(student));

}

//if roll number not found

**if**(!flag)

cout<<"Record not found.."<<**endl**;

}

file.close();

}

//=======definition of update()=================================

**void** **FileOperation::modify**(){

**int** rno,flag=0;

cout<<"Enter roll number of student:";

cin>>rno;

//open file in read mode

fstream file("StudDB.dat", ios::*in* | ios::*binary*);

//create new file for tempory data storage

fstream newFile("newFile.dat", ios::*app* |ios::*binary*);

//read records from file to search the rno

**if**(!file)

{

cout<<"File not found.."<<**endl**;

}

**else**

{

file.read((**char**\*)&student,**sizeof**(student));

**while**(!file.eof())

{

//if current record is not that to be updated

**if**(student.getRollNo()!=rno)

{

//write it to newFile

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

}

//if current record is that is to be updated

**else**

{

cout<<"Record found..."<<**endl**;

flag=1;

//update student details

string nm,add,d;

//input updated details

cout<<"Enter updated details==>"<<**endl**;

cin.ignore();

cout<<"Name:";

**getline**(cin,nm);

cout<<"Address:";

**getline**(cin,add);

cout<<"Division:";

**getline**(cin,d);

//updatting details

student.setName(nm);

student.setAddress(add);

student.setDivision(d);

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

}

//read next record

file.read((**char**\*)&student,**sizeof**(student));

}

file.close();

newFile.close();

**remove**("StudDB.dat");

**rename**("newFile.dat","StudDB.dat");

**if**(flag==0)

cout<<"Record not found..."<<**endl**;

}

}

//=========definiiton of delet()=========================================

**void** **FileOperation::delet**(){

**int** rno,flag=0;

cout<<"Enter roll number of student to be deleted:";

cin>>rno;

//open file in read mode

fstream file("StudDB.dat", ios::*in* | ios::*binary*);

//create a new file for temporary storage

fstream newFile("newFile.dat", ios::*app* | ios::*binary*);

//read records from file

file.read((**char**\*)&student,**sizeof**(student));

**if**(!file)

{

cout<<"File not found.."<<**endl**;

}

**else**{

**while**(!file.eof())

{

//if current record is not to be deleted

**if**(student.getRollNo()!=rno)

{

//write record to newFile

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

}

**else**

{

cout<<"Record deleted.."<<**endl**;

flag=1;

}

//read next record

file.read((**char**\*)&student,**sizeof**(student));

}

**if**(flag==0)

cout<<"Record not found.."<<**endl**;

file.close();

newFile.close();

**remove**("StudDB.dat");

**rename**("newFile.dat","StudDB.dat");

}

}

**FileOperation::~FileOperation**() {

// **TODO** Auto-generated destructor stub

}

* **Assignment10.cpp**

//============================================================================

// Name : Assignment10.cpp

// Author : Megha Sonavane

// Description : Student database using sequential file

//============================================================================

**#include** <iostream>

**#include**"FileOperation.h"

**using** **namespace** std;

**int** **main**() {

FileOperation fileOp;

**int** ch;

**do**{

cout<<"================MENUS================================="<<**endl**;

cout<<"1:Create new database"<<**endl**<<"2:Add new record"<<**endl**<<"3:Display all record"<<**endl**<<"4:Search a record"<<**endl**;

cout<<"5:Modify a record"<<**endl**<<"6:Delete a record"<<**endl**<<"0:Exit"<<**endl**;

cout<<"Enter choice:";

cin>>ch;

cout<<"======================================================"<<**endl**;

**switch**(ch){

**case** 1:

//create database

fileOp.create();

**break**;

**case** 2:

//adding new record

fileOp.addnew();

**break**;

**case** 3:

//display all records

fileOp.display();

**break**;

**case** 4:

//search a particular record

fileOp.search();

**break**;

**case** 5:

//modify record

fileOp.modify();

**break**;

**case** 6:

//delete record

fileOp.delet();

**break**;

**case** 0:

//exit

cout<<"Thank you...";

**break**;

**default**:

cout<<"Enter valid choice.."<<**endl**;

**break**;

}

}**while**(ch!=0);

**return** 0;

}

* **Output**

================MENUS=================================

1:Create new database

2:Add new record

3:Display all record

4:Search a record

5:Modify a record

6:Delete a record

0:Exit

Enter choice:1

======================================================

New database created...

Enter details of student:

Roll No:1001

Name:abcd

Division:se11

Address:abcd

Added successfully..

================MENUS=================================

1:Create new database

2:Add new record

3:Display all record

4:Search a record

5:Modify a record

6:Delete a record

0:Exit

Enter choice:2

======================================================

Enter details of student:

Roll No:1002

Name:pqrs

Division:pqrs

Address:pqrs

Added successfully..

================MENUS=================================

1:Create new database

2:Add new record

3:Display all record

4:Search a record

5:Modify a record

6:Delete a record

0:Exit

Enter choice:2

======================================================

Enter details of student:

Roll No:1002

Name:xyz

Division:xyz

Address:xyz

Student with roll number1002 already exist...

================MENUS=================================

1:Create new database

2:Add new record

3:Display all record

4:Search a record

5:Modify a record

6:Delete a record

0:Exit

Enter choice:3

======================================================

RollNo Name Division Address

1001 abcd se11 abcd

1002 pqrs pqrs pqrs

================MENUS=================================

1:Create new database

2:Add new record

3:Display all record

4:Search a record

5:Modify a record

6:Delete a record

0:Exit

Enter choice:5

======================================================

Enter roll number of student:1005

Record not found...

================MENUS=================================

1:Create new database

2:Add new record

3:Display all record

4:Search a record

5:Modify a record

6:Delete a record

0:Exit

Enter choice:5

======================================================

Enter roll number of student:1002

Record found...

Enter updated details==>

Name:pqrs

Address:pqrs

Division:TE4

================MENUS=================================

1:Create new database

2:Add new record

3:Display all record

4:Search a record

5:Modify a record

6:Delete a record

0:Exit

Enter choice:4

======================================================

Enter roll number to be search:1002

Record found..

RollNo Name Division Address

1002 pqrs TE4 pqrs

================MENUS=================================

1:Create new database

2:Add new record

3:Display all record

4:Search a record

5:Modify a record

6:Delete a record

0:Exit

Enter choice:3

======================================================

RollNo Name Division Address

1001 abcd se11 abcd

1002 pqrs TE4 pqrs

================MENUS=================================

1:Create new database

2:Add new record

3:Display all record

4:Search a record

5:Modify a record

6:Delete a record

0:Exit

Enter choice:2

======================================================

Enter details of student:

Roll No:1003

Name:xyz

Division:fe4

Address:xyz

Added successfully..

================MENUS=================================

1:Create new database

2:Add new record

3:Display all record

4:Search a record

5:Modify a record

6:Delete a record

0:Exit

Enter choice:3

======================================================

RollNo Name Division Address

1001 abcd se11 abcd

1002 pqrs TE4 pqrs

1003 xyz fe4 xyz

================MENUS=================================

1:Create new database

2:Add new record

3:Display all record

4:Search a record

5:Modify a record

6:Delete a record

0:Exit

Enter choice:6

======================================================

Enter roll number of student to be deleted:1005

Record not found..

================MENUS=================================

1:Create new database

2:Add new record

3:Display all record

4:Search a record

5:Modify a record

6:Delete a record

0:Exit

Enter choice:6

======================================================

Enter roll number of student to be deleted:1002

Record deleted..

================MENUS=================================

1:Create new database

2:Add new record

3:Display all record

4:Search a record

5:Modify a record

6:Delete a record

0:Exit

Enter choice:4

======================================================

Enter roll number to be search:1002

Record not found..

================MENUS=================================

1:Create new database

2:Add new record

3:Display all record

4:Search a record

5:Modify a record

6:Delete a record

0:Exit

Enter choice:3

======================================================

RollNo Name Division Address

1001 abcd se11 abcd

1003 xyz fe4 xyz

================MENUS=================================

1:Create new database

2:Add new record

3:Display all record

4:Search a record

5:Modify a record

6:Delete a record

0:Exit

Enter choice:0

======================================================

Thank you...