//Assignment no 11

#include<iostream>

#include<fstream>

#include<string.h>

using namespace std;

struct stud

{

int roll;

char name[10];

char div;

char add[10];

}rec;

class student

{

public:

void create();

void display();

int search();

void Delete();

};

void student::create()

{

char ans;

ofstream fout;

fout.open("stud.dat",ios::out|ios::binary);

do

{

cout<<"\n\tEnter Roll No of Student : ";

cin>>rec.roll;

cout<<"\n\tEnter a Name of Student : ";

cin>>rec.name;

cout<<"\n\tEnter a Division of Student : ";

cin>>rec.div;

cout<<"\n\tEnter a Address of Student : ";

cin>>rec.add;

fout.write((char \*)&rec,sizeof(stud))<<flush;

cout<<"\n\tDo You Want to Add More Records: ";

cin>>ans;

}while(ans=='y'||ans=='Y');

fout.close();

}

void student::display()

{

ifstream fin;

fin.open("stud.dat",ios::in|ios::binary);

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

cout<<"\n\tThe Content of File are:\n";

cout<<"\n\tRoll\tName\tDiv\tAddress";

while(fin.read((char \*)&rec,sizeof(stud)))

{

if(rec.roll!=-1)

cout<<"\n\t"<<rec.roll<<"\t"<<rec.name<<"\t"<<rec.div<<"\t"<<rec.add;

}

fin.close();

}

int student::search()

{

int r,i=0;

ifstream fin;

fin.open("stud.dat",ios::in|ios::binary);

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

cout<<"\n\tEnter a Roll No: ";

cin>>r;

while(fin.read((char \*)&rec,sizeof(stud)))

{

if(rec.roll==r)

{

cout<<"\n\tRecord Found...\n";

cout<<"\n\tRoll\tName\tDiv\tAddress";

cout<<"\n\t"<<rec.roll<<"\t"<<rec.name<<"\t"<<rec.div<<"\t"<<rec.add;

return i;

}

i++;

}

fin.close();

return 0;

}

void student::Delete()

{

int pos;

pos=search();

fstream f;

f.open("stud.dat",ios::in|ios::out|ios::binary);

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

if(pos==0)

{

cout<<"\n\tRecord Not Found";

return;

}

int offset=pos\*sizeof(stud);

f.seekp(offset);

rec.roll=-1;

strcpy(rec.name,"NULL");

rec.div='N';

strcpy(rec.add,"NULL");

f.write((char \*)&rec,sizeof(stud));

f.seekg(0);

f.close();

cout<<"\n\tRecord Deleted";

}

int main()

{

student obj;

int ch,key;

char ans;

do

{

cout<<"\n\t\*\*\*\*\* Student Information \*\*\*\*\*";

cout<<"\n\t1. Create\n\t2. Display\n\t3. Delete\n\t4. Search\n\t5. Exit";

cout<<"\n\t..... Enter Your Choice: ";

cin>>ch;

switch(ch)

{

case 1: obj.create();

break;

case 2: obj.display();

break;

case 3: obj.Delete();

break;

case 4: key=obj.search();

break;

case 5:

break;

}

cout<<"\n\t..... Do You Want to Continue in Main Menu: "; cin>>ans;

}while(ans=='y'||ans=='Y');

return 1;

}

output:

gescoe@gescoe-OptiPlex-3010:~/Desktop/SE-A-55$ g++ file\_11.cpp

gescoe@gescoe-OptiPlex-3010:~/Desktop/SE-A-55$ ./a.out

\*\*\*\*\* Student Information \*\*\*\*\*

1. Create

2. Display

3. Delete

4. Search

5. Exit

..... Enter Your Choice: 1

Enter Roll No of Student : 66

Enter a Name of Student : krushna

Enter a Division of Student : A

Enter a Address of Student : 812

Do You Want to Add More Records: y

Enter Roll No of Student : 55

Enter a Name of Student : Asmit

Enter a Division of Student : A

Enter a Address of Student : 50

Do You Want to Add More Records: y

Enter Roll No of Student : 47

Enter a Name of Student : Aditi

Enter a Division of Student : A

Enter a Address of Student : 11

Do You Want to Add More Records: n

..... Do You Want to Continue in Main Menu: y

\*\*\*\*\* Student Information \*\*\*\*\*

1. Create

2. Display

3. Delete

4. Search

5. Exit

..... Enter Your Choice: 2

The Content of File are:

Roll Name Div Address

66 krushna A 812

55 Asmit A 50

47 Aditi A 11

..... Do You Want to Continue in Main Menu: y

\*\*\*\*\* Student Information \*\*\*\*\*

1. Create

2. Display

3. Delete

4. Search

5. Exit

..... Enter Your Choice: 3

Enter a Roll No: 66

Record Found...

Roll Name Div Address

66 krushna A 812

Record Not Found

..... Do You Want to Continue in Main Menu: y

\*\*\*\*\* Student Information \*\*\*\*\*

1. Create

2. Display

3. Delete

4. Search

5. Exit

..... Enter Your Choice: 4

Enter a Roll No: 55

Record Found...

Roll Name Div Address

55 Asmit A 50

..... Do You Want to Continue in Main Menu: y

\*\*\*\*\* Student Information \*\*\*\*\*

1. Create

2. Display

3. Delete

4. Search

5. Exit

..... Enter Your Choice: 5

..... Do You Want to Continue in Main Menu: n

gescoe@gescoe-OptiPlex-3010:~/Desktop/SE-A-55$

### Theory:

This C++ program demonstrates the basic functionalities of managing student records using file handling in binary mode. The program allows users to perform operations like adding new student records, displaying the records, searching for a student by roll number, and deleting records. These operations are implemented using a class that defines various functions for each task. The records are stored in a binary file, "stud.dat", for persistence.

The program uses the following concepts:

1. **Binary File Operations**: The ofstream and ifstream objects are used to write and read records from a binary file.
2. **Structures**: A struct is used to define the student data, including roll number, name, division, and address.
3. **File Handling**: Operations like writing to the file (write), reading from the file (read), and manipulating file pointers using seekg and seekp.
4. **Dynamic User Interaction**: Users can repeatedly choose different options from a menu to create, display, search, or delete student records.

### Algorithm:

#### student::create() function:

1. Open the binary file "stud.dat" for writing using ofstream.
2. Enter a loop where the user is prompted to enter details of a student (roll number, name, division, and address).
3. After entering the details, write the student record to the file.
4. Ask the user whether they want to add more records. If "y" or "Y", repeat the process.
5. Close the file after the user chooses to stop.

#### student::display() function:

1. Open the binary file "stud.dat" for reading using ifstream.
2. Read the file contents using read in a loop until the end of the file.
3. For each record, print the student details (roll number, name, division, and address) to the console.
4. Close the file.

#### student::search() function:

1. Open the binary file "stud.dat" for reading using ifstream.
2. Prompt the user to enter a roll number to search for.
3. Read each student record from the file and compare the roll number with the input.
4. If a matching record is found, display the student's details and return the position.
5. If no matching record is found, return 0.

#### student::Delete() function:

1. Call the search function to find the student to be deleted.
2. If the student is found, open the file for both reading and writing (fstream).
3. Use seekp to move the file pointer to the position of the student record to be deleted.
4. Modify the student's roll number to -1 and set the name, division, and address to "NULL" to mark the record as deleted.
5. Write the modified record back to the file and close it.

#### Main Program Logic:

1. Display a menu with options for creating records, displaying records, searching records, and deleting records.
2. Based on the user's choice, call the appropriate method (create, display, search, or delete).
3. After performing an action, ask the user if they want to continue or exit the program.

### Pseudocode:

Start

Create an object of the 'student' class (obj)

Repeat until user chooses to exit:

Display Menu options

Get user's choice

Switch based on user's choice:

Case 1: Call obj.create() to add a new student record

Case 2: Call obj.display() to show all student records

Case 3: Call obj.Delete() to delete a student record

Case 4: Call obj.search() to find a student by roll number

Case 5: Exit the program

Ask the user if they want to continue

End

This program makes use of file handling in C++ to manipulate and manage student records in a binary file.