UNIVERSITY OF ENGINEERING AND TECHNOLOGY, LAHORE (NAROWAL CAMPUS)

GROUP ROLL NO. :

527,531,549

SECTION:

A

SUBJECT:

OBJECT ORIENTED PROGRAMMING(OOP)

SUBMITTED TO:

MAM SADIA TARIQ

TOPIC:

STUDENT REPORT CARD SYSTEM

STUDENT REPORT CARD SYSTEM USING OOP IN C++

OVERVIEW:

Student report card system project in C++ is a simple console application built without the use of graphics. In this project, users can perform typical report card related functions like adding a new student record and displaying, modifying, editing and deleting it. File handling has been effectively used to perform all these. This project will teach you how to use file handling in C++, add, read, display, search, modify and delete record from file.

In this code we will use classes, file handling, functions, switch statements, loop etc.

1. Create student report card record:

This feature creates a new student record containing his marks. For this the information to be provided are the name and roll no. of the student, and the marks obtained by him/her in 5 subjects – Physics, Chemistry, Math, English and Computer Science.

2. Read all students report card record:

The *void display all ()* function in this student report card system project in C++ has been used for this feature. It basically shows the progress report of all the students added in file. This feature displays the roll no. and name of all the students, the marks obtained by them in 5 subjects – Physics, Chemistry, Math, English and Computer Science, along with the percentage and grade of each student.

3. Read specific student's report card record:

This feature is same as the one explained above, except it shows the progress report and relevant data related to a particular student.

4. Modify student's report card record:

In student report card system project in C++, this feature is used to edit the report card record of a particular student. For this, the name and roll no. of the student is sought. Upon successful modification, the program displays the message "Record Updated". If no record of student is found in file, it displays the message "Record not found".

5. Delete student record:

This feature deletes the report card record of a particular student; it first of all asks for the name and roll no. of the student whose record is to be deleted.

Summary:

This C++ Student Report Card Project is a simple console application without graphics. In Student Report Card C++, users may add, display, change, edit, and delete student records.

We'll construct Student Record System in C++ using simple code. We'll build, search for, display, remove, and amend student results.

The student outcome record system will have more features. Let's examine this RRS project's characteristics. C++ Grades All of these utilize file handling. This project will show you how to utilize file handling in C++, classes, add, read, display, search, change, and remove records.

STUDENT REPORT CARD CREATED BY GROUP-02

```
#include<iostream>
#include<fstream>
#include<iomanip>
using namespace std;

class student
{
protected:
      int rollno;
      char name[50];
      double p_marks, c_marks, m_marks, e_marks, cs_marks;
      double per;
      char grade;
      void calculate();//THIS IS PRIVATE FUNCTION WHICH IS USED TO
CALCULATE GARDE
public:
      void getdata();
      void showdata() const;
      int retrollno() const;
};

void student::calculate()
{
      per = (p_marks + c_marks + m_marks + e_marks + cs_marks) / 500
*100;
      if (per >= 90)
            grade = 'A';
      else if (per >= 80)
            grade = 'B';
      else if (per >= 80)
            grade = 'C';
      else if (per >= 50)
```

```
            grade = 'D';
      else
            grade = 'F';
}

void student::getdata()
{
      cout<<"\nEnter The roll number of student : ";
      cin>>rollno;
      cout<<"\n\nEnter The Name of student : ";
      cin.ignore();
      cin.getline(name,50);
      cout<<"\nEnter The marks in physics out of 100 : ";
      cin>>p_marks;
      cout<<"\nEnter The marks in chemistry out of 100 : ";
      cin>>c_marks;
      cout<<"\nEnter The marks in maths out of 100 : ";
      cin>>m_marks;
      cout<<"\nEnter The marks in english out of 100 : ";
      cin>>e_marks;
      cout<<"\nEnter The marks in computer science out of 100 : ";
      cin>>cs_marks;
      calculate();
}

void student::showdata() const
{
      cout<<"\n\tRoll number of student : "<<rollno;
      cout<<"\n\tName of student : "<<name;
      cout<<"\n\tMarks in Physics : "<<p_marks;
      cout<<"\n\tMarks in Chemistry : "<<c_marks;
      cout<<"\n\tMarks in Maths : "<<m_marks;
      cout<<"\n\tMarks in English : "<<e_marks;
      cout<<"\n\tMarks in Computer Science :"<<cs_marks;
      cout<<"\n\tPercentage of student is  :"<<per<<"%";
      cout<<"\n\tGrade of student is :"<<grade;
      cout << "\n\n\t================================\n";
}

int  student::retrollno() const
{
      return rollno;
}

class result:public student
{
      public:
            void result1();
            void class_result();
            void display_sp(int);
```

```
            void entry_menu();
            void write_student_data();
            void modify_student(int);
            void delete_student(int);
};

int main()
{
      result res;
      char ch;
      bool flag = true;
      while(flag)
      {
            system("cls");
            cout<<"\n\n\n\tMAIN MENU";
            cout<<"\n\n\t01. RESULT MENU";
            cout<<"\n\n\t02. ENTRY/EDIT MENU";
            cout<<"\n\n\t03. EXIT";
            cout<<"\n\n\tPlease Select Your Option (1-3) ";
            cin>>ch;
            switch(ch)
            {
                  case '1': res.result1();
                        break;
                  case '2': res.entry_menu();
                        break;
                  case '3':
                        flag = false;
                        break;
                  default :cout<<"\a";
            }
    }
    system("cls");
      return 0;
}

void result::result1()
{
      char ch;
      int rno;
      system("cls");
      cout<<"\n\n\n\tRESULT MENU";
      cout<<"\n\n\n\t1. Class Result";
      cout<<"\n\n\t2. Student Report Card";
      cout<<"\n\n\t3. Back to Main Menu";
      cout<<"\n\n\n\tEnter Choice (1-3) ";
      cin>>ch;
      system("cls");
      switch(ch)
      {
```

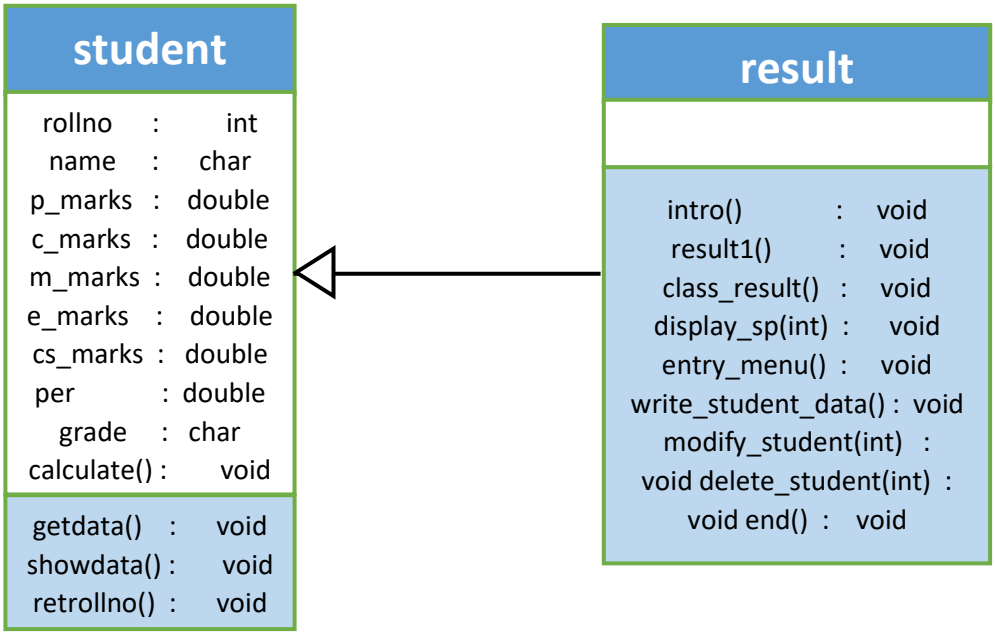
```
	case '1' :  class_result(); break;
	case '2' :  cout<<"\n\n\tEnter Roll Number Of Student : ";
cin>>rno;
                        display_sp(rno); break;
	case '3' :  break;
	default:    cout<<"\a";
	}
}

void result::class_result()
{
	result res;
	ifstream inFile;
	inFile.open("student.txt",ios::app|ios::binary);
	if(!inFile)
	{
		cout<<"File could not be open !! Press any Key...";
		cin.ignore();
		cin.get();
		return;
	}
	cout<<"\n\n\t\tALL STUDENTS RESULT \n\n";
	while(inFile.read((char *) &res, sizeof(result)))
	{
		res.showdata();
	}
	cin.ignore();
	cin.get();
	inFile.close();
}

void result::display_sp(int n)
{
	result res;
	ifstream inFile;
	inFile.open("student.txt",ios::app|ios::binary);
	if(!inFile)
	{
		cout<<"File could not be open !! Press any Key...";
		cin.ignore();
		cin.get();
		return;
	}
	bool flag=false;
	while(inFile.read((char *)&res, sizeof(result)))
	{
		if(res.retrollno()==n)
		{
			res.showdata();
			flag=true;
```

```
            }
      }
      inFile.close();
      if(flag==false)
            cout<<"\n\nrecord not exist";
      cin.ignore();
      cin.get();
}

void result::entry_menu()
{
      char ch;
      int num;
      system("cls");
      bool flag =true;
      while(flag){
      cout<<"\n\n\n\tENTRY MENU";
      cout<<"\n\n\t1.CREATE STUDENT RECORD";
      cout<<"\n\n\t2.MODIFY STUDENT RECORD";
      cout<<"\n\n\t3.DELETE STUDENT RECORD";
      cout<<"\n\n\t4.BACK TO MAIN MENU";
      cout<<"\n\n\tPlease Enter Your Choice (1-4) ";
      cin>>ch;
      system("cls");
      switch(ch)
      {
      case '1':   write_student_data(); break;
      case '2':   cout<<"\n\n\tPlease Enter The roll number ";
cin>>num;
                  modify_student(num);break;
      case '3':   cout<<"\n\n\tPlease Enter The roll number ";
cin>>num;
                  delete_student(num);break;
      case '4':   flag=false; break;
      default:    cout<<"\a";
      }
}
}

void result::write_student_data()
{
      result res;
      ofstream outFile;
      outFile.open("student.txt",ios::app|ios::binary);
      res.getdata();
      outFile.write((char *)&res, sizeof(result));
      outFile.close();
    cout<<"\n\nStudent record Has Been Created ";
      cin.ignore();
      cin.get();
```

```
	system("cls");
}

void result::modify_student(int n){
	result res;
	ifstream inFile;
	inFile.open("student.txt");
	if(!inFile)
	{
		cout<<"File could not be open !! Press any Key...";
		cin.ignore();
		cin.get();
		return;
	}
	ofstream outFile;
	outFile.open("Temp.txt");
	inFile.seekg(0,ios::beg);
	while(inFile.read((char *) &res, sizeof(result)))
	{
		if(res.retrollno()!=n)
		{
			outFile.write((char *) &res, sizeof(result));
			cout << "Record Not Exist ..";
			break;

		}
		else{
			res.getdata();
			outFile.write((char *) &res, sizeof(result));
			cout<<"\n\n\tRecord Updated ..";
		}
	}
	outFile.close();
	inFile.close();
	remove("student.txt");
	rename("Temp.txt","student.txt");
	cin.ignore();
	cin.get();
	system("cls");
}

void result::delete_student(int n)
{
	result res;
	ifstream inFile;
	inFile.open("student.txt");
	if(!inFile)
	{
		cout<<"File could not be open !! Press any Key...";
		cin.ignore();
```

```
            cin.get();
            return;
      }
      ofstream outFile;
      outFile.open("Temp.txt");
      inFile.seekg(0,ios::beg);
      while(inFile.read((char *) &res, sizeof(result)))
      {
            if(res.retrollno()!=n)
            {
                  outFile.write((char *) &res, sizeof(result));
                  cout << "Record Not Exist ..";
                  break;
            }
            else
            {
                  cout<<"\n\n\tRecord Deleted ..";
            }
      }
      outFile.close();
      inFile.close();
      remove("student.txt");
      rename("Temp.txt","student.txt");
      cin.ignore();
      cin.get();
      system("cls");
}
```

CLASS DIAGRAM:



<----------END---------->