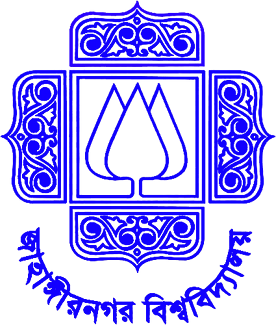
****

**PROJECT**

**STUDY TRACK MANAGEMENT SYSTEM FOR A STUDENT**

**SUBMITTED BY :: RUMANA ISLAM**

**CLASS ROLL :: 1718**

**SEMESTER :: 2-1**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**SUPERVISOR’S NAME :: Dr. MD. EZHARUL ISLAM**

**ASSISTANT PROFESSOR**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**PROJECT FEATURES**

The name of my project is “Study Track Management System for a Student”, which is a simple application in c++.

In this project,

* users can perform typical study track related functions like adding a new student academic information and displaying, modifying, editing his/her study record.

The source code is complete, totally error-free and around 800 lines. I haven’t included the source code for copy/paste here as it is a bit long;

The key features of Study Track Management System for a Student are::

* **Create a new Student Academic Information** :

This feature creates a new student Academic Information containing his name, class roll, email no, department name and semester in a text file.

* **Display Student’s Academic Information:**

It basically shows the Academic Information of student’s added in a file.

* **Create Course Information :**

In this section, there are more three options which are ::

* **Create Course credit**
* **Create tutorial marks**
* **Create Assignment marks**
* **Display Course Information :**

In this section, there are also more three options which are

* **Display Course Credit**
* **Display Tutorial Marks**
* **Display Assignment Marks**
* **Create Total Class Performance :**

This feature also Creates some another options like -

**Create Student Record :**

* creates the percentage mark of three courses
* calculates grade .

**Display Student Record :**

* It basically shows the progress report of the students added in file. This feature displays the roll no. of the students, the marks obtained by them in 3 courses– along with the percentage and grade of student.

**Modify Student Record :**

this feature is used to edit the total performance of a particular student. For this, the 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”.

**Project** **FEATURES**

* **Create Time Tracking :**

This feature creates starting and ending time to track the reading period of the student.

* **Show Time Tracking :**

It basically shows the reading time period of a student ,and this feature is very helpful for a student .

**Project** **FEATURES**

* **Header Files Used :**

Study Track Management system is a very simple project that runs with just six header files. Here, are the header files required for this project:

#include<iostream>

#include<fstream>

#include<string.h>

#include<conio.h>

#include<stdlib.h>

#include<iomanip>

**OBJECT ORIENTED PROGRAMMING FEATUERS**

**Compulsory Features that is used in my project :**

* Class, Object, Constructors
* Inheritance
* File Streams
* Multi file Programs
* Array , String
* Virtual Function

**Optional Features that is used in my project :**

* Pointer

|  |
| --- |
| **StudyInfo** |
| **-credit : integer**  **-tutorial : integer**  **-assignment : integer** |
| **+ getStudyInfo (): void**  **+displayStudyInfo (): void** |

* Operator overloading

**UML : CLASS DIAGRAM**

* **AGGREGATION : Class Within Class :**
* **INHERITANCE**

With attributes and member functions ::

|  |
| --- |
| **Student(derived)** |
| **+semester: integer**  **+roll : integer**  **+department : String** |
|  |

|  |
| --- |
| **Person(base)** |
| **+name: string**  **+email string** |
|  |

**Aggregation->**

|  |
| --- |
| **Course** |
|  |
| **+getCourseInfo(): void**  **+displayCourseInfo() :void** |

|  |
| --- |
| **Tutorial(derived)** |
| **-marks: integer**  **-courseTutorial: double** |
| **+getData():void**  **+displayData(): void** |

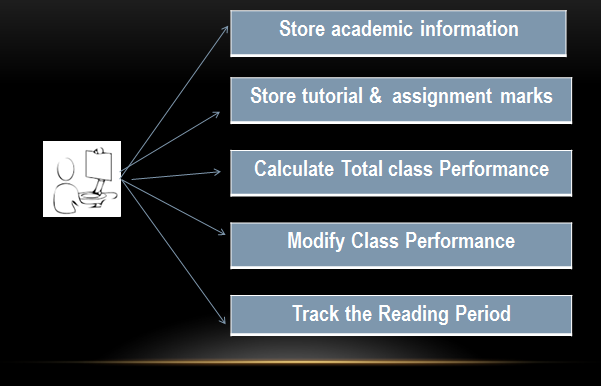
|  |
| --- |
| **Test(base)** |
| **#course : integer** |
| **+virtual getData() :void**  **+virtual displayData() :void** |

|  |
| --- |
| **Assignment(derived)** |
| **-marks: integer**  **-courseAssignment :double** |
| **+getData() :void**  **+displayData():void** |

|  |
| --- |
| **TotalClassPerformance** |
| **-rollno : integer**  **-per : float**  **-calculate : void**  **-grade : char** |
| **+getdata(): void**  **+showdata(): void**  **+retrollno() : int** |

|  |
| --- |
| **TimeTrack** |
| **-hour : integer**  **-minute : integer**  **-second : integer** |
| **+getTime(): void**  **+showTime() : void** |

**uml : Use case diagram ::**



SOURCE CODE ::

#include<iostream>

#include<fstream>

#include<string.h>

#include<conio.h>

#include<stdlib.h>

#include<iomanip>

using namespace std;

class Person

{

public:

string name,email;

};

class Student:public Person

{

public:

int semester,roll;

string department;

};

class StudyInfo

{

private:

int credit,tutorial,assignment;

public:

StudyInfo()

{

credit=0;

tutorial=0;

assignment=0;

}

StudyInfo(int a,int b,int c)

{

credit=a;

tutorial=b;

assignment=c;

}

void getStudyInfo()

{

cout<<"Enter Credit no:: ";

cin>>credit;

cout<<"Enter number of Tutorial taken :: ";

cin>>tutorial;

cout<<"Enter number of Assignment given :: ";

cin>>assignment;

}

void displayStudyInfo()

{

cout<<"Credit :: "<<credit<<endl;

cout<<"Number of Tutorial taken: "<<tutorial<<endl;

if(tutorial<1)

{

cout<<endl;

cout<<"No Tutorial has taken yet"<<endl<<endl;

}

cout<<"Number of Assignment given: "<<assignment<<endl;

if(assignment<1)

{

cout<<endl;

cout<<"No Assignment has given yet"<<endl<<endl;

}

}

};

class Course

{

private:

StudyInfo c1[3];

int i;

public:

void getCourseInfo()

{

for(int i=0; i<3; i++)

{

cout<<"Enter Data of Course "<<i+1<< " ::"<<endl<<endl;

c1[i].getStudyInfo();

}

}

void displayCourseInfo()

{

for(int i=0; i<3; i++)

{

cout<<"Data of Course "<<i+1<< " ::"<<endl<<endl;

c1[i].displayStudyInfo() ;

cout<<endl;

}

}

};

class Test

{

private:

int course[3];

public:

virtual void getData()=0;

virtual void displayData()=0;

};

class Tutorial:public Test

{

private:

int marks[3];

double courseTutorial[3][3];

public:

void getData()

{

cout<<endl;

for(int m=0; m<3; m++)

for(int c=0; c<3; c++)

{

cout<<"Enter Tutorial "<<m+1;

cout<<",Course :: "<<c+1<<"::";

cin>>courseTutorial[m][c];

}

cout<<endl;

}

void displayData()

{

cout<<"\n\n";

cout<<"\t\t COURSE\n";

cout<<"\t \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;

cout<<"\t 1\t 2\t 3"<<endl;

cout<<"\t -----------------------------------"<<endl;

for(int m=0; m<3; m++)

{

cout<<"\nTutorial "<<m+1;

for(int c=0; c<3; c++)

cout<<setiosflags(ios::fixed)<<setiosflags(ios::showpoint)<<setprecision(2)<<setw(10)<<courseTutorial[m][c];

}

cout<<endl;

cout<<endl;

}

};

class Assignment:public Test

{

private:

int marks[2];

double courseAssignment[2][3];

public:

void getData()

{

cout<<endl;

for(int j=0; j<2; j++)

for(int k=0; k<3; k++)

{

cout<<"Enter Assignment "<<j+1;

cout<<",Course :: "<<k+1<<"::";

cin>>courseAssignment[j][k];

}

cout<<endl;

}

void displayData()

{

cout<<"\n\n";

cout<<"\t\t COURSE\n";

cout<<"\t \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;

cout<<"\t 1\t 2\t 3"<<endl;

cout<<"\t -----------------------------------"<<endl;

for(int j=0; j<2; j++)

{

cout<<"\nAssignment "<<j+1;

for(int k=0; k<3; k++)

cout<<setiosflags(ios::fixed)<<setiosflags(ios::showpoint)<<setprecision(2)<<setw(10)<<courseAssignment[j][k];

}

cout<<endl;

cout<<endl;

}

};

class TotalClassPerformance

{

private:

int rollno;

int marks1, marks2, marks3;

float per;

char grade;

void calculate();

public:

void getdata();

void showdata();

void showTabular();

int retrollno();

};

void TotalClassPerformance::calculate()

{

per=(marks1+marks2+marks3)/3.0;

if(per>=60)

grade='A';

else if(per>=50)

{

grade='B';

}

else if(per>=33)

{

grade='C';

}

else

grade='F';

}

void TotalClassPerformance::getdata()

{

cout<<"\nEnter Roll no : ";

cin>>rollno;

cout<<"\nEnter The marks in Course 1 out of 100 : ";

cin>>marks1;

cout<<"\nEnter The marks in Course 2 out of 100 : ";

cin>>marks2;

cout<<"\nEnter The marks in Course 3 out of 100 : ";

cin>>marks3;

calculate();

}

void TotalClassPerformance::showdata()

{

cout<<"\nRoll number of student : "<<rollno;

cout<<"\nMarks in Course 1 : "<<marks1;

cout<<"\nMarks in Course 2 : "<<marks2;

cout<<"\nMarks in Course 3 : "<<marks3;

cout<<"\nPercentage of student is :"<<per;

cout<<"\nGrade of student is :"<<grade;

}

void TotalClassPerformance::showTabular()

{

cout<<rollno<<setw(6)<<" "<<setw(10)<<marks1<<setw(4)<<marks3<<setw(4)<<marks3<<setw(6)

<<per<<setw(6)<<" "<<grade<<endl;

}

int TotalClassPerformance::retrollno()

{

return rollno;

}

class TimeTrack

{

private:

int hour, minute,second;

public:

TimeTrack()

{

hour=0;

minute=0;

second=0;

}

TimeTrack(int x, int y,int z)

{

hour=x;

minute=y;

second=z;

}

void getTime()

{

cout<<endl;

cin>>hour>>minute>>second;

}

void displayTime()

{

cout<<endl;

cout<<hour<< " : "<<minute<<" : "<<second<<" ";

}

TimeTrack operator - (TimeTrack t)

{

TimeTrack temp;

temp.second= second- t.second;

temp.minute= minute- t.minute;

temp.hour = hour- t.hour;

if( temp.second > 59 )

{

temp.second += 60;

temp.minute--;

}

if( temp.minute > 59 )

{

temp.minute += 60;

temp.hour--;

}

if(temp.hour>12)

{

temp.hour= temp.hour +12;

}

return temp;

}

};

void Intro();

void write\_student();

void display\_all();

void modify\_student(int);

void PerformanceEntry\_menu();

int main()

{

Student \*p[100];

Test \*te[100];

fstream file;

int Count=0;

int option,n;

int i,choice;

char yn;

Assignment a;

Course c;

Tutorial t;

TimeTrack t1,t2,t3;

TotalClassPerformance cp;

Intro();

do

{

cout<<endl<<endl<<endl;

cout<<"\t\t STUDY TRACK MANAGEMENT SYSTEM FOR A STUDENT "<<endl;

cout<<"\t\t -------------------------------------"<<endl;

cout<<"\t\t MAIN MENU "<<endl;

cout<<"\t\t -------------------------------------"<<endl<<endl;

cout<<" 1.ENTER DATA of New student's ACADEMIC INFORMATION :"<<endl<<endl;

cout<<" 2.DISPLAY the ACADEMIC INFORMATION in a File :"<<endl<<endl;

cout<<" 3.TAKE Input for COURSE INFORMATION :"<<endl<<endl;

cout<<" 4.SHOW COURSE INFORMATION :"<<endl<<endl;

cout<<" 5.SHOW TOTAL CLASS PERFORMANCE :"<<endl<<endl;

cout<<" 6.ENTER TIME :"<<endl<<endl;

cout<<" 7.SHOW TIME TRACKING :"<<endl<<endl;

cout<<" Enter Your Choice :: ";

cin>>choice;

switch(choice)

{

case 1:

do

{

file.open("STUDENTT.txt", ios:: in | ios:: app | ios:: out );

p[Count]=new Student;

cout<<endl;

cout<<"Student Information: "<<endl<<endl;

cout<<" Enter Name:: ";

cin>>p[Count]->name;

cout<<" Enter Email:: ";

cin>>p[Count]->email;

cout<<" Enter department:: ";

cin>>p[Count]->department;

cout<<" Enter Semester:: ";

cin>>p[Count]->semester;

cout<<" Enter Roll no:: ";

cin>>p[Count]->roll;

file<<p[Count]->name<<endl<<p[Count]->email<<endl<<p[Count]->department<<endl<<p[Count]->semester<<endl<<p[Count]->roll<<endl<<endl;

Count++;

file.close();

cout<<endl;

cout<<"Enter another Student's info ?:(y/n):";

cin>>yn;

}

while(yn== 'y');

break;

case 2:

{

Count=0;

file.open("STUDENTT.txt", ios::in | ios::out | ios::app );

while(!file.eof())

{

p[Count]=new Student;

file>>p[Count]->name>>p[Count]->email>>p[Count]->department>>p[Count]->semester>>p[Count]->roll;

Count++;

}

for(i=0; i<Count-1; i++)

{

cout<<endl;

cout<<"Student :: "<<i+1<<endl<<endl;

cout<<" Name :: "<<p[i]->name<<endl;

cout<<" Email :: "<<p[i]->email<<endl;

cout<<" Department :: "<<p[i]->department<<endl;

cout<<" Semester :: "<<p[i]->semester<<endl;

cout<<" Roll no :: "<<p[i]->roll<<endl<<endl;

}

file.close();

break;

}

case 3:

{

cout<<endl;

cout<<"COURSE INFORMATION ::";

cout<<endl<<endl;

do

{

cout<<"\t 1.Enter COURSE CREDIT ::"<<endl<<endl;

cout<<"\t 2.Enter TUTORIAL MARKS ::"<<endl<<endl;

cout<<"\t 3.Enter ASSIGNMENT MARKS ::"<<endl<<endl;

cout<<"\t 4.BACK TO MAIN MENUE "<<endl<<endl;

cout<<" Enter Option::";

cout<<endl<<endl;

cin>>option;

cout<<endl;

switch(option)

{

case 1:

c.getCourseInfo();

break;

case 2:

te[0]=&t;

te[0]->getData();

break;

case 3:

te[1]=&a;

te[1]->getData();

break;

}

}

while(option==3);

break;

}

case 4:

{

cout<<endl;

cout<<"COURSE INFORMATION ::";

cout<<endl<<endl;

cout<<"\t 1.COURSE CREDIT ::"<<endl<<endl;

cout<<"\t 2.TUTORIAL MARKS ::"<<endl<<endl;

cout<<"\t 3.ASSIGNMENT MARKS ::"<<endl<<endl;

cout<<"\t 4.BACK TO MAIN MENU "<<endl<<endl;

cout<<" Enter Option:: ";

cin>>option;

cout<<endl;

if(option==1)

c.displayCourseInfo();

else if(option==2)

t.displayData();

else if(option==3)

a.displayData();

break;

}

case 5:

cout<<endl;

PerformanceEntry\_menu();

break;

case 6:

{

do

{

cout<<endl;

cout<<" Press 1 to Start Reading time"<<endl;

cout<<" Press 2 to End Reading time"<<endl;

cout<<" press 3 to Back to Main Menu"<<endl<<endl;

cout<<"Enter Option::";

cin>>option;

cout<<endl;

switch(option)

{

case 1:

cout<<"Starting Time:: "<<endl;

t1.getTime();

break;

case 2:

cout<<"Ending Time :: "<<endl;

t2.getTime();

break;

}

}

while(option==2);

break;

}

case 7:

{

cout<<endl;

cout<<"Starting Time is :";

cout<<endl;

t1.displayTime();

cout<<endl;

cout<<"Ending Time is :";

cout<<endl;

t2.displayTime();

cout<<endl;

cout<<endl;

t3=t2-t1;

cout<<"Your Reading Period :: ";

cout<<endl;

t3.displayTime();

cout<<endl<<endl;

}

default:

cout<<endl;

cout<<"\n";

break;

}

}

while(choice!=8);

return 0;

}

void Intro()

{

cout<<"\n\n\t\t\t\tPROJECT";

cout<<"\n\n\n\t\t STUDY TRACK MANAGEMENT SYSTEM FOR A STUDENT";

cout<<"\n\n\n\t\t MADE BY\t :: RUMANA ISLAM";

cout<<"\n\n\t\t SEMESTER NO\t :: 2-1";

cout<<"\n\n\t\t ROLL NO\t :: 1718";

cout<<endl;

cout<<"\n\n\t\t SUPERVISOR'S NAME:: Dr.MD. EZHARUL ISLAM";

cout<<"\n\n\t\t ASSISTANT PROFESSOR";

cout<<"\n\n\t\t DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING"<<endl;

getch();

}

void write\_student()

{

TotalClassPerformance cp;

ofstream outFile;

outFile.open("TTTT.txt", ios::in | ios::out | ios::app);

cp.getdata();

outFile.write((char \*) &cp, sizeof(TotalClassPerformance));

outFile.close();

cout<<"\n\nStudent record Has Been Created ";

getch();

}

void display\_all()

{

TotalClassPerformance cp;

ifstream inFile;

inFile.open("TTTT.txt", ios::in | ios::out | ios::app);

if(!inFile)

{

cout<<"File could not be open !! Press any Key...";

getch();

return;

}

cout<<"\n\n\n\t\tDISPLAY RECORD !!!\n\n";

while(inFile.read((char \*) &cp, sizeof(TotalClassPerformance)))

{

cp.showdata();

}

inFile.close();

getch();

}

void modify\_student(int n)

{

int found=0;

TotalClassPerformance cp;

fstream File;

File.open("TTTT.txt", ios::in | ios::out | ios::app);

if(!File)

{

cout<<"File could not be open !! Press any Key...";

getch();

return;

}

while(File.read((char \*) &cp, sizeof(TotalClassPerformance)) && found==0)

{

if(cp.retrollno()==n)

{

cp.showdata();

cout<<"\n\nPlease Enter The New Details of student"<<endl;

cp.getdata();

int pos=(-1)\*sizeof(cp);

File.seekp(pos,ios::cur);

File.write((char \*) &cp, sizeof(TotalClassPerformance));

cout<<"\n\n\t Record Updated";

found=1;

}

}

File.close();

if(found==0)

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

getch();

}

void PerformanceEntry\_menu()

{

char ch;

int num;

cout<<"\n\n\n\tENTRY MENU";

cout<<"\n\n\t1.CREATE STUDENT RECORD";

cout<<"\n\n\t2.DISPLAY STUDENT'S RECORDS";

cout<<"\n\n\t3.MODIFY STUDENT RECORD";

cout<<"\n\n\t4.BACK TO MAIN MENU";

cout<<"\n\n\tPlease Enter Your Choice (1-4) ";

cin>>ch;

switch(ch)

{

case '1':

write\_student();

break;

case '2':

display\_all();

break;

case '3':

cout<<"\n\n\tPlease Enter The roll number ";

cin>>num;

modify\_student(num);

break;

case '4':

break;

default:

cout<<"\n";

PerformanceEntry\_menu();

}

}

