#include<iostream>

#include<string>

#include<vector>

#include<fstream>

using namespace std;

//grade class to add all info about grades

class Grade {

private:

double grade;

public:

// adding data to grade

void addData() {

cin >> grade;

}

double getgradesforAVG() {

return grade;

}

void display() {

cout << grade << endl;

}

string getDataForFile() {

return to\_string(grade) + " ";

}

void loadDataFromFile(ifstream& file) {

string GradeSTR;

getline(file, GradeSTR);

grade = stod(GradeSTR);

}

};

// grade to add new student

class Student {

private:

string Name;

string Email;

string StudentID;

double average = 0;

double FourthGrade;

vector<Grade>grades; //store a list of grades

Grade mark; // declaring an object to connect the classes Student and Grade

double total = 0;

public:

void addGrades()//add grade for user

{

mark.addData();

grades.push\_back(mark);//adding to vector

}

void addData()

{

cout << "enter name: " << endl;

getline(cin >> ws, Name);

cout << "enter Email: " << endl;

getline(cin >> ws, Email);

cout << "enter student number: " << endl;

getline(cin >> ws, StudentID);

/\*cin.ignore(1000, '\n');

cin.clear();

getline(cin, StudentID);\*/

}

void findAVG()

{

total = 0; // resets the total

for (int i = 0; i < grades.size(); i++) {

double marking = grades[i].getgradesforAVG();

total += marking;

}

int count = grades.size();

average = total / count;

}

void CalculateFourthGrade()

{

FourthGrade = 280 - total;

}

//function to return the data

string getName() {

return Name;

}

string getEmail() {

return Email;

}

string getStudenId() {

return StudentID;

}

vector<Grade> getGrades() {

return grades;

}

string getAVG()

{

return to\_string(average) + " ";

}

string getFourthGrade()

{

return to\_string(FourthGrade) + " ";

}

void displayAVG() {

findAVG();

cout << average << endl;

}

void displayFourthGrade() {

CalculateFourthGrade();

cout << "The Grade " << Name << " needs to achieve in the fourth assessment to get average of 70% is: " << FourthGrade << endl;

}

void printGrades() {

for (int i = 0; i < grades.size(); i++) {

cout << "Grade " << (i + 1) << ": ";

grades[i].display();

}

}

void display()

{

cout << "Student name: " << Name << endl;

cout << "student number: " << StudentID << endl;

cout << "email: " << Email << endl;

printGrades();

findAVG();

cout << "average is: " << average << endl;

}

void edit() //editing function

{

//maybe add a eature that the person canchoose to change only a part of it not go through each step

cout << "enter new name: " << endl;

getline(cin >> ws , Name);

cout << "enter new email: " << endl;

getline(cin >> ws, Email);

cout << "enter new student number: " << endl;

getline(cin >> ws, StudentID);

}

void LoadDataFromFile(ifstream& file)

{

getline(file, Name);

getline(file, Email);

getline(file, StudentID);

string averageSTR;

getline(file, averageSTR);

/\*when average was written to the file it was converted to string so when we load it we need to get the variable as a string and the convert it to double\*/

average = stod(averageSTR);

//getting te grades from the file

bool MoreGrade;

Grade markobj;

while (MoreGrade = true) {

cout << "i dont want to " << endl;

markobj.loadDataFromFile(file); //calling the function from the grade class

grades.push\_back(markobj); // adding new grade

streampos oldpos = file.tellg(); // gives the current location in the file

string checkEnding;// dedicating a variable to check wether it has reached the end of file or not

getline(file, checkEnding);

if (checkEnding == "LastLineYay") {

MoreGrade = false;

break;

}

else {

file.seekg(oldpos); // continues getting the new info from that position until it reaches the "lastlineyay"

}

}

}

};

// writing the menu

class Menu {

private:

int option;

int numberInput;

vector<Student> students; // holds every student

bool choose = true;

//printing a studets record

void printstudentvector()

{

for (int i = 0; i < students.size(); i++)

{

cout << (i + 1) << " " << students[i].getName() << endl;

}

}

//add new student function

void addNewStudent()

{

Student Students; //calling an object

Students.addData();

students.push\_back(Students); // adding the new student to the vector

}

//add new grade for a student

void addNewGrade()

{

cout << "which student you ant to add anew grade for ? " << endl;

printstudentvector();

cin >> numberInput;

students[numberInput - 1].addGrades();

}

void findingAverage() {

cout << "which student average would you like to see? " << endl;

printstudentvector();

cin >> numberInput;

students[numberInput - 1].displayAVG();

}

void findingFourthGrade() {

cout << "which student average would you like to see? " << endl;

printstudentvector();

cin >> numberInput;

students[numberInput - 1].displayFourthGrade();

}

void displayStudent()

{

cout << "which student would u like to see?" << endl;

printstudentvector(); // printing the list of students

cin >> numberInput;

students[numberInput - 1].display();

}

void EditStudent()

{

cout << "which student would you like to edi? " << endl;

printstudentvector();

cin >> numberInput;

students[numberInput - 1].edit(); // calling the edit function in the student class for a specific student in the vector

}

void WriteToFile()

{

ofstream outputFile; // creating the fileobject

outputFile.open("studentmarks.txt"); //opening the file

for (int i = 0; i < students.size(); i++)

{

outputFile << students[i].getName() << endl;

outputFile << students[i].getEmail() << endl;

outputFile << students[i].getStudenId() << endl;

outputFile << students[i].getAVG() << endl;

for (auto y : students[i].getGrades())

{

outputFile << y.getDataForFile() << endl;

}

outputFile << "LastLineYay"; // seperating each student info by seting checkpoint

if (i != (students.size() - 1))

{

outputFile << endl;

}

}

}

void loadFromFile()

{

Student studentobj;

ifstream inputFile;

inputFile.open("studentmarks.txt"); //opening file

if (inputFile)

{

while (!inputFile.eof())

{

studentobj.LoadDataFromFile(inputFile); //loading data from the file

students.push\_back(studentobj); //add the students to the vector

}

}

}

void saveAndQuit()

{

WriteToFile();

choose = false;

cout << "Goodbye!" << endl;

}

public:

void start() {

while (choose) {

cout << "|----------------------------------------------------------------------|" << endl;

cout << "|Choose from the menu provided below: |" << endl;

//showing the menu to the user

cout << "|----------------------------------------------------------------------|" << endl;

cout << "|1. Add a new studnt |" << endl;

cout << "|2. Add a student grade |" << endl;

cout << "|3. Load the students list and show a student's information |" << endl;

cout << "|4. Edit the list |" << endl;

cout << "|5. Calculate Average |" << endl;

cout << "|6. predict grade |" << endl;

cout << "|7. Save data to file |" << endl;

cout << "|8. load from file |" << endl;

cout << "|9. Save the data and quit |" << endl;

cout << "|----------------------------------------------------------------------|" << endl;

cin >> option;

//using while loop to validate the user's input

while (option < 1 || option > 9) {

cout << "invalid input! try again" << endl;

cin >> option;

}

switch (option) {

case 1:

addNewStudent();

break;

case 2:

addNewGrade();

break;

case 3:

displayStudent();

break;

case 4:

EditStudent();

break;

case 5:

findingAverage();

break;

case 6 :

findingFourthGrade();

break;

case 7:

WriteToFile();

break;

case 8:

loadFromFile();

break;

case 9:

saveAndQuit();

break;

}

}

}

};

int main() {

//getting the menu class and from there each class will be called in each setion without calling them in the main

Menu menu;

menu.start();

}