```cpp
//Brian Andersen
//This programme creates and manipulates a gradebook.

#include <iostream>
#include <iomanip>
#include <windows.h>
#include <fstream>


using namespace std;

struct student {
    string name;
    int year;
    int grades[5];
    double gpa;
};
//-----------------------------------
void init(student roster[], int SIZE);
int setup(student roster[]);
void printall(student roster[], int numstus);
void sortbyname(student roster[], int numstus);
void sortbygpa(student roster[], int numstus);
void highestgpainclass(student roster[], int numstus, int year);
int addstudent(student roster[], int numstus, int SIZE);
int menu();
void writetofile(student roster[], int numstus);
int readfromfile(student roster[], int SIZE);
int deletestu(student roster[], int numstus);
void printone(student roster[], int numstus);
void update(student roster[], int numstus);
//-----------------------------------
int main()
{
    const int SIZE = 31;
    student roster[SIZE];
    int numstus, ch, year;

    init(roster, SIZE);
    //numstus = setup(roster);
    numstus = readfromfile(roster, SIZE);

    do {
        ch = menu();
        switch(ch) {
            case 1: printall(roster, numstus);
                    break;
            case 2: sortbyname(roster, numstus);
                    printall(roster, numstus);
                    break;
            case 3: sortbygpa(roster, numstus);
                    printall(roster, numstus);
                    break;
            case 4: cout << "Enter a year: ";
                    cin >> year;
                    highestgpainclass(roster, numstus, year);
                    break;
            case 5: numstus = addstudent(roster, numstus, SIZE);
                    break;
            case 6: numstus = deletestu(roster, numstus);
```

```cpp
                    break;
            case 7: printone(roster, numstus);
                    break;
            case 8: update(roster, numstus);
                    break;
            case 0: writetofile(roster, numstus);
                    break;
        }
    } while (ch !=0);

    return 0;
}
//-----------------------------------
int menu()
{
    int x;

    do {
        cout << "Enter 1 to print all students and their info." << endl;
        cout << "Enter 2 to sort students by name and print." << endl;
        cout << "Enter 3 to sort students by GPA and print." << endl;
        cout << "Enter 4 to print the student with the highest GPA." << endl;
        cout << "Enter 5 to add a student to the roster." << endl;
        cout << "Enter 6 to remove a student from the roster." << endl;
        cout << "Enter 7 to print a singular student's info." << endl;
        cout << "Enter 8 to update a student's info." << endl;
        cout << "Enter 0 to save and quit." << endl;
        cout << endl;
        cout << "Enter your choice: ";
        cin >> x;
        if (x < 0 || x > 8) {
            cout << "Enter a value between 0 and 7." << endl;
        }
        cout << endl;
    } while (x < 0 || x > 8);
    cout << endl;
    return x;
}
//-----------------------------------
void init(student roster[], int SIZE)
{
    int i;

    for (i = 0; i < SIZE; i++)
        roster[i] = {"", 0, {0,0,0,0,0}, 0.0};
}
//-----------------------------------
int setup(student roster[])
{
    roster[0] = {"Nichols, Van", 2020, {85,85,91,89,92}, 88.4};
    roster[1] = {"Carr, Edward", 2021, {92,79,88,94,89}, 88.4};
    roster[2] = {"Willis, Lois", 2023, {87,85,97,82,94}, 89.0};
    roster[3] = {"Byrd, Norma", 2022, {66,69,78,84,82}, 75.8};
    roster[4] = {"Schwartz, Alejandro", 2020, {84,88,90,94,88}, 88.8};
    roster[5] = {"Beck, Alicia", 2020, {77,87,92,84,94}, 86.8};
    roster[6] = {"Evans, Cheryl", 2021, {80,77,86,74,78}, 79.0};
    roster[7] = {"Harrington, Sylvester", 2022, {79,81,87,83,90}, 84.0};
    roster[8] = {"Jones, Dolores", 2021, {82,87,80,89,89}, 85.4};
    roster[9] = {"Aguilar, Myrtle", 2023, {69,73,77,67,80}, 73.2};
```

```cpp
        return 10;
}
//-----------------------------------
void printall(student roster[], int numstus)
{
    int i, j;

    system("CLS");

    for (i = 0; i < numstus; i++) {
        cout << left << setw(22) << roster[i].name
             << right << setw(5) << roster[i].year;
        for (j = 0; j < 5; j++)
            cout << right << setw(10) << roster[i].grades[j];
        cout << right << setw(15) << fixed << setprecision(1) << roster[i].gpa <<
endl;
    }
    cout << endl;
}
//-----------------------------------
void sortbyname(student roster[], int numstus)
{
    int i, j;
    student temp;

    for (i = 0; i < numstus - 1; i++)
        for (j = i + 1; j < numstus; j++)
            if (roster[i].name > roster[j].name) {
                temp = roster[i];
                roster[i] = roster[j];
                roster[j] = temp;
            }
    printall(roster, numstus);
}
//-----------------------------------
void sortbygpa(student roster[], int numstus)
{
int i, j;
student temp;

    for (i = 0; i < numstus - 1; i++)
        for (j = i + 1; j < numstus; j++)
            if (roster[i].gpa > roster[j].gpa) {
                temp = roster[i];
                roster[i] = roster[j];
                roster[j] = temp;
            }
    printall(roster, numstus);
}
//-----------------------------------
void highestgpainclass(student roster[], int numstus, int year)
{
    int i, j;
    student highest;
    highest.gpa = -1;

    for (i = 0; i < numstus; i++)
        if (roster[i].year == year && roster[i].gpa > highest.gpa)
            highest = roster[i];
```

```cpp
    cout << left << setw(22) << highest.name << right << setw(5) << highest.year <<
right;
    for (j = 0; j < 5; j++)
        cout << right << setw(10) << highest.grades[j] << right << setw(10)
            << right << setw(15) << fixed << setprecision(1) << highest.gpa << endl
<< endl;
}
//-----------------------------------
int addstudent(student roster[], int numstus, int SIZE)
{
    int year, i, grade, gradtot;
    double gpa;
    gradtot = 0;

    if (numstus == SIZE)
        cout << "Your roster is out of room." << endl;
    else {
        cout << "Please enter the student's name: ";
        cin.ignore();
        getline(cin, roster[numstus].name);
        do {
            cout << "Enter the student's year: ";
            cin >> year;
            if (year < 2020 || year > 2023)
                cout << "Enter a year between 2020 and 2023." << endl;
        } while (year < 2020 || year > 2023);
        roster[numstus].year = year;
        for (i = 0; i < 5; i++) {
            do {
                cout << "Enter grade #" << i + 1 << ": ";
                cin >> grade;
                if (grade < 0 || grade > 100)
                    cout << "Enter a grade between 0 and 100." << endl;
            } while (grade < 0 || grade > 100);
            roster[numstus].grades[i] = grade;
            gradtot += grade;
        }
        gpa = gradtot / 5.0;
        roster[numstus].gpa = gpa;
        numstus++;
        return numstus;
    }
}
//-----------------------------------
void writetofile(student roster[], int numstus)
{
    string filename;
    int i, j;
    ofstream outfile;

    cout << endl;
    cout << "Enter Filename: ";
    cin >> filename;

    outfile.open (filename, ios::out);

    if (!outfile) {
        cerr << "Cannot open file " << filename << "." << endl;
        exit(1);
    }
```

```cpp
    else {
        sortbyname (roster, numstus);

        outfile << numstus << endl;

        for (i = 0; i < numstus; i++) {
            outfile << roster[i].name << "/";

            outfile << roster[i].year << "/";

            for (j = 0; j < 5; j++)
                outfile << roster[i].grades[j] << "/";

            outfile << roster[i].gpa << endl;
        }
    }
    outfile.close();
}
//---------------------------------
int readfromfile(student roster[], int SIZE)
{
    string filename, s;
    int i, j, numstus;
    ifstream infile;

    cout << "Enter filename: ";
    cin >> filename;
    infile.open(filename, ios::in);

    if (!infile) {
        cerr << "Error: Number of students exceeds class size." << endl;
        exit(2);
    }
    else {
        getline (infile, s);

        numstus = atoi(s.c_str());

        if (numstus > SIZE) {
            cerr << "Error: Number of students exceeds class size." << endl;
            exit(3);
        }
    }

    for (i = 0; i < numstus; i++) {
            getline(infile, s, '/');
            roster[i].name = s;

            getline(infile, s, '/');
            roster[i].year = atoi(s.c_str());

            for (j = 0; j < 5; j++) {
                getline(infile, s, '/');
                roster[i].grades[j] = atoi(s.c_str());
            }

            getline(infile, s, '\n');
            roster[i].gpa = atof (s.c_str());
    }
    infile.close();
```

```cpp
        return numstus;
}
//----------------------------------
int deletestu(student roster[], int numstus)
{
    int i, j;
    string name;



    cout << "Enter name of student: ";
    cin.ignore();
    getline(cin, name);

    i = 0;
    while (i < numstus && roster[i].name != name) {
        i++;
    }

    if (i < numstus) {
        for (j = i; j < numstus - 1; j++)
            roster[j] = roster[j + 1];
        roster[numstus - 1] = {"", 0, {0,0,0,0,0}, 0.0};
        numstus--;
    }
    else
        cout << "This student was not found. Please try again." << endl << endl;
    return numstus;
}
//----------------------------------
void printone(student roster[], int numstus) {
    int i, j;
    string name;

    cout << "Enter name of student: ";
    cin.ignore();
    getline(cin, name);
    cout << endl;

    i = 0;
    while (i < numstus && roster[i].name != name) {
        i++;
    }

    if (i < numstus) {
        cout << left << setw(22) << roster[i].name
             << right << setw(5) << roster[i].year;
        for (j = 0; j < 5; j++)
            cout << right << setw(10) << roster[i].grades[j];
        cout << right << setw(15) << fixed << setprecision(1) << roster[i].gpa <<
endl << endl;
    }
    else
        cout << "This student was not found. Please try again." << endl << endl;
}
//----------------------------------
void update(student roster[], int numstus) {
    int i, j, gradtot, x, y, z;
    string name;
```

```cpp
printall(roster, numstus);
cout << endl;
cout << "Enter the student's name: ";
cin.ignore();
getline(cin, name);
cout << endl;

i = 0;
while (i < numstus && roster[i].name != name) {
    i++;
}
if (i < numstus) {
    cout << "Enter 1 to change the student's graduation year." << endl;
    cout << "Enter 2 to change one of the student's grades." << endl;
    cout << "Enter your choice now: ";
    cin >> x;

    do {
        cout << "Enter 1 to change the student's graduation year." << endl;
        cout << "Enter 2 to change one of the student's grades." << endl;
        cout << endl;
        cout << "Enter your choice now: ";
        cin >> x;
        if (x < 1 || x > 2)
            cout << "Enter a value between 1 and 2." << endl;
        cout << endl;
    } while (x < 1 || x > 2);

    switch(x) {
        case 1: do {
                    cout << "Enter the student's new graduation year: ";
                    cin >> y;
                    if (y < 2020 || y > 2023)
                        cout << "Enter a value between 2020 and 2023. " << endl;
                    cout << endl;
                } while (y < 2020 || y > 2023);

                roster[i].year = y;
                break;
        case 2: do {
                    cout << "Which grade would you like to alter? (1-5): ";
                    cin >> y;
                    if (y < 1 || y > 5)
                        cout << "Enter a value between 1 and 5." << endl;
                    cout << endl;
                } while (y < 1 || y > 5);

                do {
                    cout << "Enter the student's new grade: ";
                    cin >> z;
                    if (z < 0 || z > 100)
                        cout << "Enter a value between 0 and 100." << endl;
                } while (z < 0 || z > 100);

                roster[i].grades[y-1] = z;

                gradtot = 0;
                for (j = 0; j < 5; j++) {
                    gradtot += roster[i].grades[j];
```

```cpp
                }

                roster[i].gpa = gradtot / 5.0;
                break;
            }
            printall(roster, numstus);
        }
        else
            cout << "This student was not found. Please try again." << endl << endl;
}
```