**LAB 11 – SUBSCRIPT OPERATOR**

#include <iostream>

#include <string>

#include <stdexcept>

using namespace std;

class Student {

private:

string name;

int rollNo;

float marks[3];

float average;

string grade;

bool feesPaid;

static int studentCount;

public:

Student(string name = "Unknown", int rollNo = 0, bool feesPaid = false) {

this->name = name;

this->rollNo = rollNo;

this->feesPaid = feesPaid;

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

marks[i] = 0.0;

}

this->average = 0.0;

this->grade = "F";

studentCount++;

}

Student(string name, int rollNo, float mark1, float mark2, float mark3, bool feesPaid = false) {

this->name = name;

this->rollNo = rollNo;

this->marks[0] = mark1;

this->marks[1] = mark2;

this->marks[2] = mark3;

this->average = 0.0;

this->grade = "F";

this->feesPaid = feesPaid;

studentCount++;

}

Student(const Student &student) {

name = student.name;

rollNo = student.rollNo;

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

marks[i] = student.marks[i];

}

average = student.average;

grade = student.grade;

feesPaid = student.feesPaid;

studentCount++;

}

static int getStudentCount() {

return studentCount;

}

void calculateAverage() {

float total = 0;

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

total += marks[i];

}

average = total / 3;

}

void calculateGrade() {

if (average >= 90) {

grade = "A+";

} else if (average >= 80) {

grade = "A";

} else if (average >= 75) {

grade = "B+";

} else if (average >= 70) {

grade = "B";

} else if (average >= 60) {

grade = "C";

} else {

grade = "F";

}

}

void payFees() {

this->feesPaid = true;

cout << name << " has paid the fees." << endl;

}

void displayDetails(bool showMarks, bool showGrade, bool showFeesStatus) const {

cout << "\nStudent Name: " << name << endl;

cout << "Roll Number: " << rollNo << endl;

if (showMarks) {

cout << "Marks in 3 subjects:";

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

cout << " " << marks[i];

}

cout << endl;

}

if (showGrade) {

cout << "Average Marks: " << average << endl;

cout << "Grade: " << grade << endl;

}

if (showFeesStatus) {

cout << "Fees Paid: " << (feesPaid ? "Yes" : "No") << endl;

}

}

bool isEligibleForExams() const {

return feesPaid && (average > 60);

}

float& operator[](int index) {

if (index < 0 || index >= 3) {

throw out\_of\_range("Index out of range.");

}

return marks[index];

}

Student& operator++() {

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

marks[i]++;

}

calculateAverage();

calculateGrade();

return \*this;

}

Student& operator--() {

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

marks[i]--;

}

calculateAverage();

calculateGrade();

return \*this;

}

Student operator-() {

Student temp = \*this;

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

temp.marks[i] = -temp.marks[i];

}

temp.calculateAverage();

temp.calculateGrade();

return temp;

}

Student operator+(const Student &s) {

Student temp = \*this;

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

temp.marks[i] += s.marks[i];

}

temp.calculateAverage();

temp.calculateGrade();

return temp;

}

Student operator-(const Student &s) {

Student temp = \*this;

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

temp.marks[i] -= s.marks[i];

}

temp.calculateAverage();

temp.calculateGrade();

return temp;

}

bool operator==(const Student &s) {

return (this->average == s.average);

}

friend void compareStudents(const Student &s1, const Student &s2);

string getName() const { return name; } // Getter for name

};

int Student::studentCount = 0;

void compareStudents(const Student &s1, const Student &s2) {

cout << "\nComparing students " << s1.getName() << " and " << s2.getName() << ":\n";

if (s1.average > s2.average) {

cout << s1.getName() << " has higher marks with an average of " << s1.average << endl;

} else if (s2.average > s1.average) {

cout << s2.getName() << " has higher marks with an average of " << s2.average << endl;

} else {

cout << "Both have the same average marks.\n";

}

}

int main() {

Student students[3] = {

Student("John", 101, false),

Student("Bob", 102, false),

Student("Alice", 103, true)

};

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

students[i].calculateAverage();

students[i].calculateGrade();

}

students[0].payFees();

students[1].payFees();

students[1][0] = 80.0;

students[2][1] = 75.0;

cout << "\nFull Information of Students (Including Fees Status):";

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

students[i].displayDetails(true, true, true);

}

cout << "\nExam Eligibility Status:\n";

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

cout << students[i].getName() << (students[i].isEligibleForExams() ? " is eligible for exams." : " is not eligible for exams.") << endl;

}

compareStudents(students[1], students[2]);

cout << "\nTotal number of students: " << Student::getStudentCount() << endl;

return 0;

}