VIT - Vellore

Name: RONIT MEXSON .
Email: ronit.mexson2024@vitstudent.ac.in
Roll no: 24BAI0036
Phone: 9999999999
Branch: ARUMUGA ARUN R_OOPS
Department: admin
Batch: VL2024250502365
Degree: admin

Scan to verify results

BCSE102P_Structured and Object Oriented Programming Lab_VL2024250502365

VIT V_Structured and OOP_Lab 7_COD_Easy_Operator Overloading

Attempt : 1
Total Mark : 20
Marks Obtained : 20

Section 1 : Coding

1. Problem Statement

Implement a Fraction class that represents a fraction with a numerator and a denominator. Overload the '+' operator to add two fractions and return the result as a simplified fraction.

Function Specifications: Fraction operator+(const Fraction&amp; other) const

Answer

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

```
int gcd(int a, int b) {
    while (b != 0) {
        int temp = b;
        b = a % b;
        a = temp;
    }
    return a;
}

class Fraction {
public:
    int numerator;
    int denominator;

    Fraction(int num, int den) {
        numerator = num;
        denominator = den;
    }

    void simplify() {
        int g = gcd(numerator, denominator);
        numerator /= g;
        denominator /= g;
    }

    Fraction operator+(const Fraction& other) const {
        int num = numerator * other.denominator + other.numerator * denominator;
        int den = denominator * other.denominator;
        Fraction result(num, den);
        result.simplify();
        return result;
    }

    void display() {
        cout << numerator << "/" << denominator << endl;
    }
};

int main() {
    int num1, den1, num2, den2;
    cin >> num1 >> den1;
    cin >> num2 >> den2;
    Fraction f1(num1, den1);
    Fraction f2(num2, den2);
    Fraction result = f1 + f2;

    result.display();

    return 0;
}
```

Status : Correct ***Marks : 10/10***

2. Problem Statement

John is driving a car with an initial velocity (in m/s) that suddenly accelerates at a constant rate (in m/s2) for a certain time (in seconds). He wants to write a program that calculates and displays the final velocity of the car.

Help John calculate the final velocity by overloading the * operator in the Acceleration class.

Formula: Final velocity = Initial velocity + (Acceleration * time)

Answer

```
// You are using GCC
#include <iostream>
#include <iomanip>
using namespace std;

class Acceleration {
    float acc;

public:
    Acceleration(float a) {
        acc = a;
    }

    float operator*(float time) {
        return acc * time;
    }
};

int main() {
    float initialVelocity, acceleration, time;
    cin >> initialVelocity >> acceleration >> time;

    Acceleration a(acceleration);

    float finalVelocity = initialVelocity + (a * time);

    cout << fixed << setprecision(1);
    cout << finalVelocity << " m/s" << endl;

    return 0;
}
```

Status : Correct ***Marks : 10/10***