**NAME**- SANDIPAN CHAKRABORTY

**STREAM**- COMPUTER SCIENCE AND BUSINESS SYSTEM

**YEAR**- SECOND (FOURTH SEMESTER)

**ROLL NUMBER**- 18731122015

**SUBJECT NAME**- OOP WITH C++ LAB

**SUBJECT CODE-** PCC CSBS 492

**REGISTRATION NO**- 221870110094

**SESSION-** 2022-2026

**LAB ASSIGNMENT**

**1. Create a class** **Vector** **that will represent a vector [Ai+Bj+Ck format]. Overload  unary - operator so that they    operate on the objects of** **Vector.  Also develop this with friend function.**

**Input: 10 20 30**

**Output: (-10)i + (-20)j +(-30)k**

* **CODE: -**

#include <iostream>

{

using namespace std;

class Vector {

public:

int A, B, C;

Vector(int a, int b, int c) : A(a), B(b), C(c) {}

Vector operator-() {

return Vector(-A, -B, -C);

}

friend Vector operator-(const Vector& vec) {

return Vector(-vec.A, -vec.B, -vec.C);

}

};

int main() {

Vector v1(1, 2, 3);

Vector v2 = -v1;

Vector v3 = operator-(v1);

cout << "Original Vector: " << v1.A << "i + " << v1.B << "j + " << v1.C << "k" <<endl;

cout << "Unary Minus Result (Overloaded): " << v2.A << "i + " << v2.B << "j + " << v2.C << "k" << endl;

cout << "Unary Minus Result (Friend Function): " << v3.A << "i + " << v3.B << "j + " << v3.C << "k" <<endl;

return 0;

}

**OUTPUT**

ORIGINAL VECTOR: - 10i + -20j + - 30k

PRIMARY MINUS RESULT (Overloaded): -10i + - 20j + - 30k

PRIMARY MINUS RESULT (Friend Function): -10i + -20j + -30k

**2. Create a class COMPLEX that will represent a complex number [A+iB format]. Overload arithmetic + and arithmetic - operators so that they operate on the objects of COMPLEX. Also develop this with friend function**

**Input:**

**line1: 5 4**

**line2: 6 -3**

**Output: 11+i**

**CODE: -**

#include <iostream>

using namespace std;

class COMPLEX {

private:

float real;

float image;

public:

COMPLEX(float r, float i) : real(r), image(i) {}

COMPLEX operator+(const COMPLEX& c) {

return COMPLEX(real + c.real, image + c.image);

}

COMPLEX operator-(const COMPLEX& c) {

return COMPLEX(real - c.real, image - c.image);

}

friend COMPLEX operator+(const COMPLEX& c1, const COMPLEX& c2);

friend COMPLEX operator-(const COMPLEX& c1, const COMPLEX& c2);

void display() {

cout << real << " + " << image << "i" << endl;

}

};

COMPLEX operator+(const COMPLEX& c1, const COMPLEX& c2) {

return COMPLEX(c1.real + c2.real, c1.image + c2.image);

}

COMPLEX operator-(const COMPLEX& c1, const COMPLEX& c2) {

return COMPLEX(c1.real - c2.real, c1.image - c2.image);

}

int main() {

COMPLEX num1(3, 4);

COMPLEX num2(1, 2);

COMPLEX sum = num1 + num2;

COMPLEX diff = num1 - num2;

sum.display();

diff.display();

return 0;

}

**OUTPUT**

4+6i

2+2i

**3. Create a time class that will contain:hour,minute and second. Create two time class object and add them in such a way that if second is greater than 60 it will be 1 minute and if minute is greater than 60 it will be 1 hour.**

**->CODE: -**

#include <iostream>

class Time {

private:

int hour;

int minute;

int second;

public:

Time(int h, int m, int s) : hour(h), minute(m), second(s) {

Time operator+(const Time& t) {

int h = hour + t.hour;

int m = minute + t.minute;

int s = second + t.second;

if (s >= 60) {

m += s / 60;

s %= 60;

}

if (m >= 60) {

h += m / 60;

m %= 60;

}

return Time(h, m, s);

}

void displayTime() {

std::cout << "Time: " << hour << " hours, " << minute << " minutes, " << second << " seconds" << std::endl;

}

};

int main() {

Time t1(1, 70, 30);

Time t2(2, 30, 45);

Time sum = t1 + t2;

sum.displayTime();return 0;

}

**OUTPUT**

Time: 4 hours, 41 minutes, 15 seconds