## operator overloading(12-08-24)

Author: ThanhTH10

Date: 12/08/2024

1. write a program to overload unary operator for processing counters.It should support both upward and downward counting.It must also support operator for adding two counters and storing the result in another counter.

#include <iostream>

using namespace std;

class Counter

{

    int count;

public:

    Counter(int c) : count(c) {}

    Counter operator++();

    Counter operator++(int);

    Counter operator--();

    Counter operator--(int);

    Counter operator+(const Counter &c);

    int getCount() const { return count; }

};

Counter Counter::operator++()

{

    count++;

    return \*this;

}

Counter Counter::operator++(int)

{

    Counter temp = \*this;

    count++;

    return temp;

}

Counter Counter::operator--()

{

    count--;

    return \*this;

}

Counter Counter::operator--(int)

{

    Counter temp = \*this;

    count--;

    return temp;

}

Counter Counter::operator+(const Counter &c)

{

    return Counter(count + c.count);

}

int main(int argc, char const \*argv[])

{

    Counter c1(10);

    Counter c2(20);

    cout << "Initial count of c1: " << c1.getCount() << endl;

    cout << "Initial count of c2: " << c2.getCount() << endl;

    // Upward counting

    Counter c3 = ++c1;

    cout << "Count of c1 after increment: " << c1.getCount() << endl;

    cout << "Count of c3: " << c3.getCount() << endl;

    // Downward counting

    Counter c4 = --c2;

    cout << "Count of c2 after decrement: " << c2.getCount() << endl;

    cout << "Count of c4: " << c4.getCount() << endl;

    // Add two counters

    Counter c5 = c1 + c2;

    cout << "Count of c5 (c1 + c2): " << c5.getCount() << endl;

    return 0;

}

2.wap to overload '+' operator in complex numbers addition using friend function.

#include <iostream>

class Complex

{

private:

    double real;

    double imag;

public:

    Complex(double r = 0, double i = 0) : real(r), imag(i) {}

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

    void print()

    {

        std::cout << real << " + " << imag << "i" << std::endl;

    }

};

Complex operator+(const Complex &c1, const Complex &c2)

{

    return Complex(c1.real + c2.real, c1.imag + c2.imag);

}

int main()

{

    Complex c1(3, 4);

    Complex c2(2, 1);

    Complex c = c1 + c2;

    c.print();

    return 0;

}