

Philip Pesic

Week 15

November 27 2022

Week 15 Prog 2

Given the following Complex Number class,

Add three more overloaded operators: $+$, $*$ and $/$.

Test.

```
/* C++ program to demonstrate the overloading of binary operator by subtracting one complex  
number from another. */
```

```
#include <iostream>
```

```
using namespace std;
```

```
class Complex
```

```
{
```

```
    private:
```

```
        float real;
```

```
        float imag;
```

```
    public:
```

```
        Complex(): real(0), imag(0){ }
```

Philip Pesic

Week 15

November 27 2022

Week 15 Prog 2

```
void input()

{

cout<<"Enter real and imaginary parts respectively: ";

cin>>real;

cin>>imag;

}

Complex operator - (Complex c2)    /* Operator Function */

{

Complex temp;

temp.real=real-c2.real;

temp.imag=imag-c2.imag;

return temp;

}

void output()

{
```

Philip Pesic

Week 15

November 27 2022

Week 15 Prog 2

```
        if(imag<0)

            cout<<"Output Complex number: "<<real<<imag<<"i";

        else

            cout<<"Output Complex number: "<<real<<"+"<<imag<<"i";

        }

};

int main()

{

    Complex c1, c2, result;

    cout<<"Enter first complex number:\n";

    c1.input();

    cout<<"Enter second complex number:\n";

    c2.input();

    /* In case of operator overloading of binary operators in C++ programming, the object on right
    hand side of operator is always assumed as argument by compiler. */
```

Philip Pesic

Week 15

November 27 2022

Week 15 Prog 2

```
        result=c1-c2; /* c2 is furnished as an argument to the operator function. */

        result.output();

        return 0;

    }

//

// main.cpp
// Week 15 Prog 2
//

// Created by Pippo Pesic on 11/25/22.
//

#include <iostream>

using namespace std;

class Complex
{
    private:
        float real;
```

Philip Pesic

Week 15

November 27 2022

Week 15 Prog 2

```
float imag;

public:

Complex(): real(0), imag(0){ }

void input()

{

cout<<"Enter real and imaginary parts respectively: ";

cin>>real;

cin>>imag;

}

Complex operator - (Complex c2)    /* Operator Function */

{

Complex temp;

temp.real=real-c2.real;

temp.imag=imag-c2.imag;

return temp;

}

Complex operator + (Complex a2)    /* Operator Function */

{

Complex temp;

temp.real=real+a2.real;

temp.imag=imag+a2.imag;
```

Philip Pesic

Week 15

November 27 2022

Week 15 Prog 2

```
    return temp;
```

```
}
```

```
Complex operator * (Complex m2)  /* Operator Function */
```

```
{
```

```
    Complex temp;
```

```
    temp.real=real*m2.real;
```

```
    temp.imag=imag*m2.imag;
```

```
    return temp;
```

```
}
```

```
Complex operator / (Complex d2)  /* Operator Function */
```

```
{
```

```
    Complex temp;
```

```
    temp.real=real/d2.real;
```

```
    temp.imag=imag/d2.imag;
```

```
    return temp;
```

```
}
```

```
void output()
```

Philip Pesic

Week 15

November 27 2022

Week 15 Prog 2

```
{  
    if(imag<0)  
        cout<<"Output Complex number: "<<real<<imag<<"i" << endl;  
    else  
        cout<<"Output Complex number: "<<real<<"+"<<imag<<"i" << endl;  
}  
};  
  
int main()  
{  
    Complex c1, c2, a1, a2, m1, m2, d1, d2, result;  
    cout<<"Enter first complex number:\n";  
    c1.input();  
    cout<<"Enter second complex number:\n";  
    c2.input();  
    result=c1-c2; /* c2 is furnished as an argument to the operator function. */  
    result.output();  
  
    cout<<"Enter first complex number:\n";  
    a1.input();  
    cout<<"Enter second complex number:\n";  
    a2.input();
```

Philip Pesic

Week 15

November 27 2022

Week 15 Prog 2

```
result=a1+a2; /* c2 is furnished as an argument to the operator function. */
```

```
result.output();
```

```
cout<<"Enter first complex number:\n";
```

```
m1.input();
```

```
cout<<"Enter second complex number:\n";
```

```
m2.input();
```

```
result=m1*m2; /* c2 is furnished as an argument to the operator function. */
```

```
result.output();
```

```
cout<<"Enter first complex number:\n";
```

```
d1.input();
```

```
cout<<"Enter second complex number:\n";
```

```
d2.input();
```

```
result=a1/d2; /* c2 is furnished as an argument to the operator function. */
```

```
result.output();
```

```
/* In case of operator overloading of binary operators in C++ programming, the object on right  
hand side of operator is always assumed as argument by compiler. */
```

```
cout << "Philip Pesic 11/27/22" << endl;
```

```
return 0;
```


Philip Pesic

Week 15

November 27 2022

Week 15 Prog 2

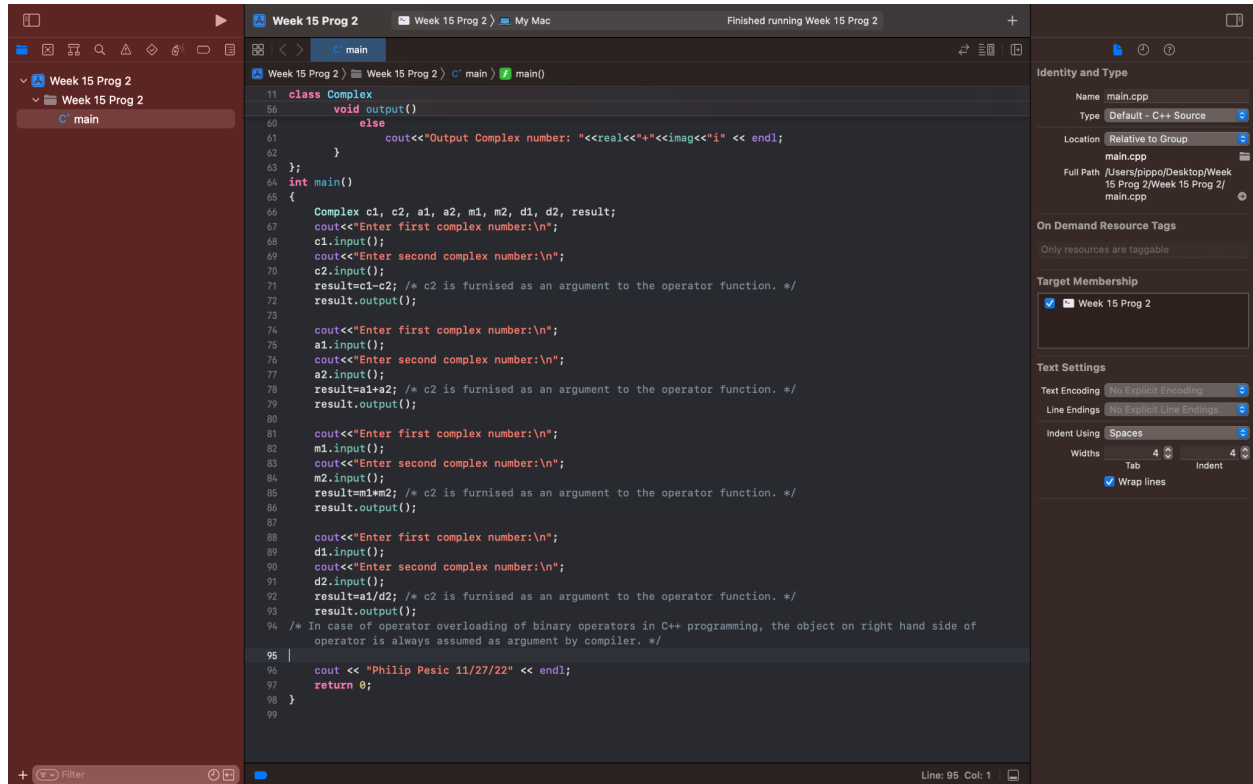
}

Philip Pesic

Week 15

November 27 2022

Week 15 Prog 2



Philip Pesic

Week 15

November 27 2022

Week 15 Prog 2

The screenshot shows a C++ IDE with a project named "Week 15 Prog 2". The main.cpp file contains the following code:

```
11 class Complex
12 {
13     public:
14     Complex operator - (Complex c2) /* Operator Function */
15     {
16         Complex temp;
17         temp.real=real-c2.real;
18         temp.imag=imag-c2.imag;
19         return temp;
20     }
21     Complex operator + (Complex a2) /* Operator Function */
22     {
23         Complex temp;
24         temp.real=real+a2.real;
25         temp.imag=imag+a2.imag;
26         return temp;
27     }
28     Complex operator * (Complex m2) /* Operator Function */
29     {
30         Complex temp;
31         temp.real=real*m2.real;
32         temp.imag=imag*m2.imag;
33         return temp;
34     }
35     Complex operator / (Complex d2) /* Operator Function */
36     {
37         Complex temp;
38         temp.real=real/d2.real;
39         temp.imag=imag/d2.imag;
40         return temp;
41     }
42     void output()
43     {
44         if(imag<0)
45             cout<<"Output Complex number: "<<real<<imag<<"i" << endl;
46         else
47             cout<<"Output Complex number: "<<real<<"+ "<<imag<<"i" << endl;
48     }
49 };
50 int main()
51 {
52     Complex c1, c2, a1, a2, m1, m2, d1, d2, result;
53     cout<<"Enter first complex number:\n";
```

The IDE interface includes a sidebar on the left with a file explorer showing the project structure. On the right, there is a "Properties" panel for the selected file, showing details like "Name: main.cpp", "Type: Default - C++ Source", and "Location: Relative to Group". Below this, there are sections for "On Demand Resource Tags", "Target Membership" (with a checkbox for "Week 15 Prog 2"), and "Text Settings" (including options for Text Encoding, Line Endings, Indent Using, Widths, and Wrap lines).

Philip Pesic

Week 15

November 27 2022

Week 15 Prog 2

The screenshot shows a C++ IDE with a project named "Week 15 Prog 2". The main.cpp file contains the following code:

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

class Complex
{
private:
    float real;
    float imag;
public:
    Complex(): real(0), imag(0){ }
    void input()
    {
        cout<<"Enter real and imaginary parts respectively: ";
        cin>>real;
        cin>>imag;
    }
};
```

The output window shows the program's execution, which prompts the user to enter real and imaginary parts and then outputs the complex number in the form $a+bi$. The output is as follows:

```
Enter first complex number:
Enter real and imaginary parts respectively: 145
154
Enter second complex number:
1Enter real and imaginary parts respectively: 41
5
14Output Complex number: 4+149i
Enter first complex number:
5Enter real and imaginary parts respectively: 14
5413
Enter second complex number:
Enter real and imaginary parts respectively: 53
15134
Output Complex number: 14567+20547i
Enter first complex number:
Enter real and imaginary parts respectively: 4
1
Enter second complex number:
Enter real and imaginary parts respectively: 145
154
Output Complex number: 588+154i
Enter first complex number:
Enter real and imaginary parts respectively: 2
4
Enter second complex number:
6Enter real and imaginary parts respectively: 7
8
Output Complex number: 216.627+676.625i
Philip Pesic 11/27/22
Program ended with exit code: 0
```

I learned: how to write my own operators