Overloading stream insertion (<>) operators in C++

Difficulty Level : Medium ● Last Updated : 16 Jun, 2021

In C++, stream insertion operator “<<” is used for output and extraction operator “>>” is used for input.

We must know the following things before we start overloading these operators.

1) cout is an object of ostream class and cin is an object of istream class

2) These operators must be overloaded as a global function. And if we want to allow them to access private data members of the class, we must make them friend.

Why these operators must be overloaded as global?

In operator overloading, if an operator is overloaded as a member, then it must be a member of the object on the left side of the operator. For example, consider the statement “ob1 + ob2” (let ob1 and ob2 be objects of two different classes). To make this statement compile, we must overload ‘+’ in a class of ‘ob1’ or make ‘+’ a global function.

The operators ‘<<‘ and ‘>>’ are called like ‘cout << ob1’ and ‘cin >> ob1’. So if we want to make them a member method, then they must be made members of ostream and istream classes, which is not a good option most of the time. Therefore, these operators are overloaded as global functions with two parameters, cout and object of user-defined class.

Following is a complete C++ program to demonstrate overloading of <> operators.

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

class Complex
{
private:
    int real, imag;
public:
    Complex(int r = 0, int i =0)
    {  real = r;   imag = i; }
    friend ostream & operator << (ostream &out, const Complex &c);
    friend istream & operator >> (istream &in,  Complex &c);
};

ostream & operator << (ostream &out, const Complex &c)
{
    out << c.real;
    out << "+i" << c.imag << endl;
    return out;
}

istream & operator >> (istream &in,  Complex &c)
{
    cout << "Enter Real Part ";
    in >> c.real;
    cout << "Enter Imaginary Part ";
    in >> c.imag;
    return in;
}

int main()
{
   Complex c1;
   cin >> c1;
   cout << "The complex object is ";
   cout << c1;
   return 0;
}
```

Output:

```
Enter Real Part 10
Enter Imaginary Part 20
The complex object is 10+i20
```

Please write comments if you find anything incorrect, or you want to share

Related Articles

1. Input/Output Operators Overloading in C++
2. Rules for operator overloading
3. Function Overloading and Return Type in C++
4. Function overloading and const keyword
5. Does overloading work with Inheritance?
6. Overloading Subscript or array index operator [] in C++
7. C++ | Operator Overloading | Question 10
8. Constructor Overloading in C++
9. Overloading function templates in C++
10. Namespaces in C++ | Set 4 (Overloading, and Exchange of Data in different Namespaces)

Previous

Default Assignment Operator and References in C++

Next

Operator Overloading in C++

Article Contributed By :

GeeksforGeeks

Vote for difficulty

Current difficulty : Medium

Improved By : jacksonthall22, 1devyansh1

Article Tags : cpp-operator-overloading, cpp-overloading, C++

Practice Tags : CPP

Improve Article Report Issue

A-143, 9th Floor, Sovereign Corporate Tower,
Sector-136, Noida, Uttar Pradesh - 201305

feedback@geeksforgeeks.org

Computer Science

GATE CS Notes

Operating Systems

Computer Network

Database Management System

Software Engineering

Digital Logic Design

Engineering Maths

Data Science & ML

Data Science With Python

Data Science For Beginner

Machine Learning Tutorial

Maths For Machine Learning

Pandas Tutorial

NumPy Tutorial

NLP Tutorial

Interview Corner

Company Preparation

Preparation for SDE

Company Interview Corner

Experienced Interview

Internship Interview

Competitive Programming

Aptitude

Python

Python Tutorial

Python Programming Examples

Django Tutorial

Python Projects

Python Tkinter

OpenCV Python Tutorial

GfG School

CBSE Notes for Class 8

CBSE Notes for Class 9

CBSE Notes for Class 10

CBSE Notes for Class 11

CBSE Notes for Class 12

English Grammar

UPSC/SSC/BANKING

SSC CGL Syllabus

SBI PO Syllabus

IBPS PO Syllabus

UPSC Ethics Notes

UPSC Economics Notes

UPSC History Notes