

Overloading insertion and extraction operator

- In C++, stream insertion operator “<<” is used for output and extraction operator “>>” is used for input.
- **cout is an object of ostream class and cin is an object of istream class**
- C++ is able to input and output the built-in data types using the stream extraction operator >> and the stream insertion operator <<.
- The stream insertion and stream extraction operators also can be overloaded to perform input and output for user-defined types like an object.
- Here, it is important to make operator overloading function a friend of the class because it would be called without creating an object.

Program

```
#include<iostream>
#include<fstream>
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
```