OPERATOR OVERLOADING USING A FRIEND FUNCTION

You can overload an operator for a class by using a nonmember function, which is usually a friend of the class. Since a friend function is not a member of the class, it does not have a this pointer. Therefore, an overloaded friend operator function is passed the operands explicitly. This means that a friend function that overloads a binary operator has two parameters, and a friend function that overloads a unary operator has one parameter. When overloading a binary operator using a friend function, the left operand is passed in the first parameter and the right operand is passed in the second parameter.

In this program, the operator+( ) function is made into a friend:

```cpp
#include <iostream>
using namespace std;
class loc {
    int longitude, latitude;
public:
    loc() {} // needed to construct temporaries
    loc(int lg, int lt) {
        longitude = lg;
        latitude = lt;
    }
    void show() {
        cout << longitude << " ";
        cout << latitude << "\n";
    }
    friend loc operator+(loc op1, loc op2); // now a friend
    loc operator-(loc op2);
    loc operator=(loc op2);
    loc operator++();
};
// Now, + is overloaded using friend function.
```
loc operator+(loc op1, loc op2)
{
    loc temp;
    temp.longitude = op1.longitude + op2.longitude;
    temp.latitude = op1.latitude + op2.latitude;
    return temp;
}
// Overload - for loc.
loc loc::operator-(loc op2)
{
    loc temp;
    // notice order of operands
    temp.longitude = longitude - op2.longitude;
    temp.latitude = latitude - op2.latitude;
    return temp;
}
// Overload assignment for loc
loc loc::operator=(loc op2)
{
    longitude = op2.longitude;
    latitude = op2.latitude;
    return *this; // i.e., return object that generated call
}
// Overload ++ for loc.
loc loc::operator++()
{
    longitude++;          
    latitude++;          
    return *this;         
}
int main()
{
loc ob1(10, 20), ob2(5, 30);
ob1 = ob1 + ob2;
ob1.show();
return 0;
}

There are some restrictions that apply to friend operator functions. First, you may not overload the =, ( ), [ ], or \rightarrow operators by using a friend function. Second, as explained in the next section, when overloading the increment or decrement operators, you will need to use a reference parameter when using a friend function.