The unary operators operate on a single operand and following are the examples of Unary operators:

- **The increment ++ and decrement -- operators.**
- The unary minus – operator.
- The logical not ! operator.

The unary operators operate on the object for which they were called and normally, this operator appears on the left side of the object, as in !obj, -obj, and ++obj but sometime they can be used as postfix as well like obj++ or obj--.

Following example explain how minus – operator can be overloaded for prefix as well as postfix usage.

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

class Distance
{
private:
    int feet;   // 0 to infinite
    int inches; // 0 to 12

public:
    // required constructors
    Distance(){
        feet = 0;
        inches = 0;
    }
    Distance(int f, int i){
        feet = f;
        inches = i;
    }
    // method to display distance
    void displayDistance()
    {
        cout << "F: " << feet << " I:" << inches << endl;
    }
    // overloaded minus (-) operator
    Distance operator- ()
    {
        feet = -feet;
        inches = -inches;
        return Distance(feet, inches);
    }
};

int main()
{
    Distance D1(11, 10), D2(-5, 11);

    -D1;       // apply negation
    D1.displayDistance(); // display D1

    -D2;       // apply negation
    D2.displayDistance(); // display D2

    return 0;
}
```

When the above code is compiled and executed, it produces the following result:
Hope above example makes your concept clear and you can apply similar concept to overload Logical Not Operators. 

Loading [MathJax]/jax/output/HTML-CSS/jax.js