```
#include <iostream>

using namespace std;

class Lab9
{
public:
    int a[2]; // could be private, but methods whould be needed to access this member.

    // operators as member functions: 'this' is bound to the left hand operand
    bool operator>(const Lab9 &x) const;
    void operator+(const Lab9 &x);
    void operator*(int factor);
    void Printvals();
    // constructor(s)
    // destructor
};

// --------------- Member functions ---------------------------------
bool Lab9::operator>(const Lab9 &x) const
{
    if ((a[0] > x.a[0]) & (a[1] > x.a[1]))
        return 1;
    else
        return 0;
}

void Lab9::operator+(const Lab9 &x)
{
    a[0] += x.a[0];
    a[1] += x.a[1];
}

void Lab9::operator*(int factor)
{
    a[0] *= factor;
    a[1] *= factor;
}

void Lab9::Printvals()
{
    cout << "\na[0] = " << a[0] << ", a[1] = " << a[1] << endl;
}

// -------- Operators can be Non-member functions -------------------
Lab9 operator+(const Lab9 &lhs, const Lab9 &rhs)
{
    Lab9 sum;
    sum.a[0] = lhs.a[0] + rhs.a[0];
    sum.a[1] = lhs.a[1] + rhs.a[1];

    return sum;
}
```

```
// Main function. Shows a few examples about using the operators.
int main()
{
    Lab9 obj1, obj2, obj3;
    int f = 10;

    obj1.a[0] = 1; obj1.a[1] = 5;
    obj2.a[0] = 10; obj2.a[1] = 20;

    if(obj1 > obj2) // normal expression. Which operator is being called?
        cout << "\nObject 1 is bigger than object 2" << endl;
    else
        cout << "\nObject 1 is not bigger than object 2" << endl;

     if(obj1.operator>(obj2))    // equivalent function call (eq. to obj1 > obj2)
        cout << "\nObject 1 is bigger than object 2" << endl;
    else
        cout << "\nObject 1 is not bigger than object 2" << endl;


    if(obj2 > obj1) // Which operator is being called?
        cout << "\nObject 2 is bigger than object 1" << endl;
    else
        cout << "\nObject 2 is not bigger than object 1" << endl;

    if(obj2.operator>(obj1))    // explicit function call, equivalent to the previous
        cout << "\nObject 2 is bigger than object 1" << endl;
    else
        cout << "\nObject 2 is not bigger than object 1" << endl;

    cout << "------------------------------------------------------------------" << endl;
    obj1.Printvals();
    obj2.Printvals();

    //obj1 + obj2;  // normal expression. Which operator is being called?
    obj1.operator+(obj2);   // equivalent, try it out
    obj1.Printvals();
    obj2.Printvals();

    //obj2 + obj1;  // normal expression. Which operator is being called?
    obj2.operator+(obj1);   // equivalent, try it out
    obj1.Printvals();
    obj2.Printvals();

    cout << "------------------------------------------------------------------" << endl;
    obj1*2;     // Would 2*obj1 work? COMPILE ERROR
    obj2*f;     // Would f*obj2 work? COMPILE ERROR
    obj1.Printvals();
    obj2.Printvals();

    cout << "------------------------------------------------------------------" << endl;
    // Which operator is being called next?
    //obj3 = operator+(obj1, obj2); // Would  obj3 = obj2 + obj1; work?
    Lab9 hold=obj2;
    (obj2+obj1);
    obj3=obj2;
    obj2=hold;
    obj1.Printvals();               // If not, how could you make it work?
    obj2.Printvals();
    obj3.Printvals();

    obj3 = operator+(obj2, obj1);   // is this the same as before?
    obj1.Printvals();
    obj2.Printvals();
    obj3.Printvals();

    return 0;
}
```