

ITEC243 – Lecture Session – 18-May-2020

Operator Overloading: Unary and binary operators

Example:

```
//test.h
class test{
private:
    int x, y;
public:
    test(int x = 0, int y = 0)
    {
        this->x = x;
        this->y = y;
    }
    void display()
    {
        cout << "(" << this->x << "," << this->y << ")";
    }
    //prefix increment
    void operator++()
    {
        ++this->x;
        ++this->y;
    }
    //postfix increment
    void operator++(int notused)
    {
        this->x += 3;
        this->y += 3;
    }

    //prefix decrement
    void operator--()
    {
        this->x -= 2;
        this->y -= 2;
    }
    //postfix decrement
    void operator--(int notused)
    {
        this->x--;
        this->y--;
    }
    //binary operator for addition , multiplication, subtraction
    friend test operator+(test&, test&);
    friend test operator*(test&, test&);
```

```

        friend test operator-(test&, test&);
};
test operator+(test& A, test& B)
{
    test temp;
    temp.x = A.x + B.x;
    temp.y = A.y + B.y;
    return temp;
}
test operator*(test& A, test& B)
{
    test temp;
    temp.x = A.x*B.x;
    temp.y = A.y*B.y;
    return temp;
}
test operator-(test& A, test& B)
{
    test temp;
    temp.x = A.x - B.y;
    temp.y = A.x - B.x;
    return temp;
}

```

//test.cpp

```

#include<iostream>
using namespace std;
#include"test.h"
void main()
{
    test obj1(4,5), obj2(5,8), obj3, obj4;
    obj1++;
    ++obj2;
    obj1.display(); cout << endl;
    obj2.display(); cout << endl;

    obj3 = obj1*obj2;
    obj1.display(); cout << "*";
    obj2.display(); cout << "=";
    obj3.display(); cout << endl;

    obj3--;
    ++obj3;

    obj4 = obj3 - obj1;
    obj3.display(); cout << "-";
    obj1.display(); cout << "=";
    obj4.display(); cout << endl;
}

```

```
system("pause");  
}
```