

**PROGRAM** – TO OVERLOAD THE POST INCREMENT AND PRE INCREMENT OPERATOR USING CLASS AND DISPLAY THE DIFFERENCE BETWEEN THEIR FUNCTIONING

**//BHAVNA VERMA - 171210019-05/02/2019**

**//OVERLOADING OF POST INCREMENT AND PRE INCREMENT OPERATOR**

```
#include<iostream>

using namespace std;

class increment
{
    int count;

public :
    //CONSTRUCTOR INITIALISING VALUE OF COUNT
    increment()
    {
        count=0;
    }

    //OVERLOADING FOR PRE INCREMENT OPERATOR
    increment operator ++ ()
    {
        increment t;
        t.count=++count;
        return t;
    }

    //OVERLOADING FOR POST INCREMENT OPERATOR
    increment operator ++ (int)
    {
        increment t;
        t.count=count++;
        return t;
    }


    void display()
    {
        printf("\nVALUE OF COUNT IS -> %d", count);
    }
};
```

```

int main()
{
    increment o1, o2;

    cout<<"\nTHE VALUE OF COUNT INITIALLY IN TWO OBJECTS AS SHOWN : "<<endl;
    cout<<"\nOBJECT 1";
    o1.display();
    cout<<"\nOBJECT 2";
    o2.display();
    cout<<"\nPOST INCREMENT IS DONE : object1 = object2++ "<<endl;
    o1=o2++;
    cout<<"\nOBJECT 1";
    o1.display();
    cout<<"\nOBJECT 2";
    o2.display();
    cout<<"\n\nIT SHOWS THAT VALUE OF OBJECT 1 IS FIRST USED AND INCREMENTED
LATER\n ";
    cout<<"\nPRE INCREMENT IS DONE : object1 = ++object2 "<<endl;
    o1=++o2;
    o1.display();
    return 0;
}

```

 C:\Users\VERMA\Downloads\increment.exe

```

THE VALUE OF COUNT INITIALLY IN TWO OBJECTS AS SHOWN :
OBJECT 1
VALUE OF COUNT IS -> 0
OBJECT 2
VALUE OF COUNT IS -> 0
POST INCREMENT IS DONE : object1 = object2++

OBJECT 1
VALUE OF COUNT IS -> 0
OBJECT 2
VALUE OF COUNT IS -> 1
"IT SHOWS THAT VALUE OF OBJECT 1 IS FIRST USED AND INCREMENTED LATER"
PRE INCREMENT IS DONE : object1 = ++object2

VALUE OF COUNT IS -> 2
-----
Process exited after 3.85 seconds with return value 0
Press any key to continue . . .

```