Overloading Special Operators

C++ has the ability to provide the operators with a special meaning for a data type, this ability is known as operator overloading. For example, we can overload an operator '+' in a class like String so that we can concatenate two strings by just using +.

Some of the special types of operators overloading in C++ are as follows:

- new – This is employed to allocate the memory dynamically.
- Delete – This is employed to free the memory dynamically.
- [] – This is employed as a subscript operator.
- -> – – This is employed as a member access operator.
- = – This is employed to assign the values.
- () – This is employed for function calls.

The operators of the types of operators overloading in C++ apart from those mentioned above could be overloaded often as a member or even as non-members. However, in most cases, non-member overloading is suggested. Because:

Symmetry: Anytime a binary operator is determined as a class process, it should have objects as its operands. We ought to compose like complex*5 but not like 5*complex since 5. operator*(complex)does not make any sense. Put simply, a*b must be the just like b*a. Or else, it breaks the cumulativeness that this end-user is planning on using the *operator. Therefore, in that instance, we must make use of no-member operators overloading.

Weak coupling: since a non-member approach can't gain access to private members, it has a tendency to create the classless coupled types of operator overloading in C++

Example of unary types of operator overloading in C++:

Using unary types of operator overloading in C++:

```
//Overload ++ when used as prefix

#include

Using namespace std;

Class count

{

Private:

Int value;

Public:
```

```
//constructor to initialize count to 5

Count() : value(5)  {}

//overload ++ when used as prefix

Void operator ++ ()

{

++value;

}

Void display()

{

Cout<<"Count: "<<value<<endl;

}

};

Int main()

{

Count count1;

//call the "void operator ++ ()" function

++count1:

Count1.display();

Return 0;

}
```

The output of unary types of operator overloading in C++:

Count: 6