Operator Overloading

To assign more than one operation on an same operator is known as Operator overloading.

Operator overloading is used to overload or redefines most of the operators available in C++.

It is used to perform the operation on the user-defined data type.

Syntax of Operator Overloading

To overload an operator, we use a special operator function.

```
returntype classname :: operator op(argument list)
{
    // body of the function
}
```

→ return type is the type of value.
→ class name is the name of the class.
→ Operator is the keyword, op is the operator being overloaded.

* Rules for Operator Overloading

- The overloaded operator contains atleast one operand of the user defined data type.
- Only existing operators can be overloaded.
- Precedence of the operators cannot be changed.
- We cannot change the basic meaning of an operator. Like '+' to subtract.
- Overloaded operator follow the syntax rules of the original operators.
- The Syntax of an operator cannot be changed.
- Some of the Operators that cannot be overloaded are:
  (i) Scope resolution operator (::)
  (ii) Class member access Operators (.)

(iii) sizeof operator
(iv) Conditional operator (?:)

- We cannot use friend functions to overload certain operators.
- Some operator like assignment "=", address "&" and comma "," are by default overloaded.
- Unary operators declared as member functions take no arguments; if declared as global functions, they take one argument.

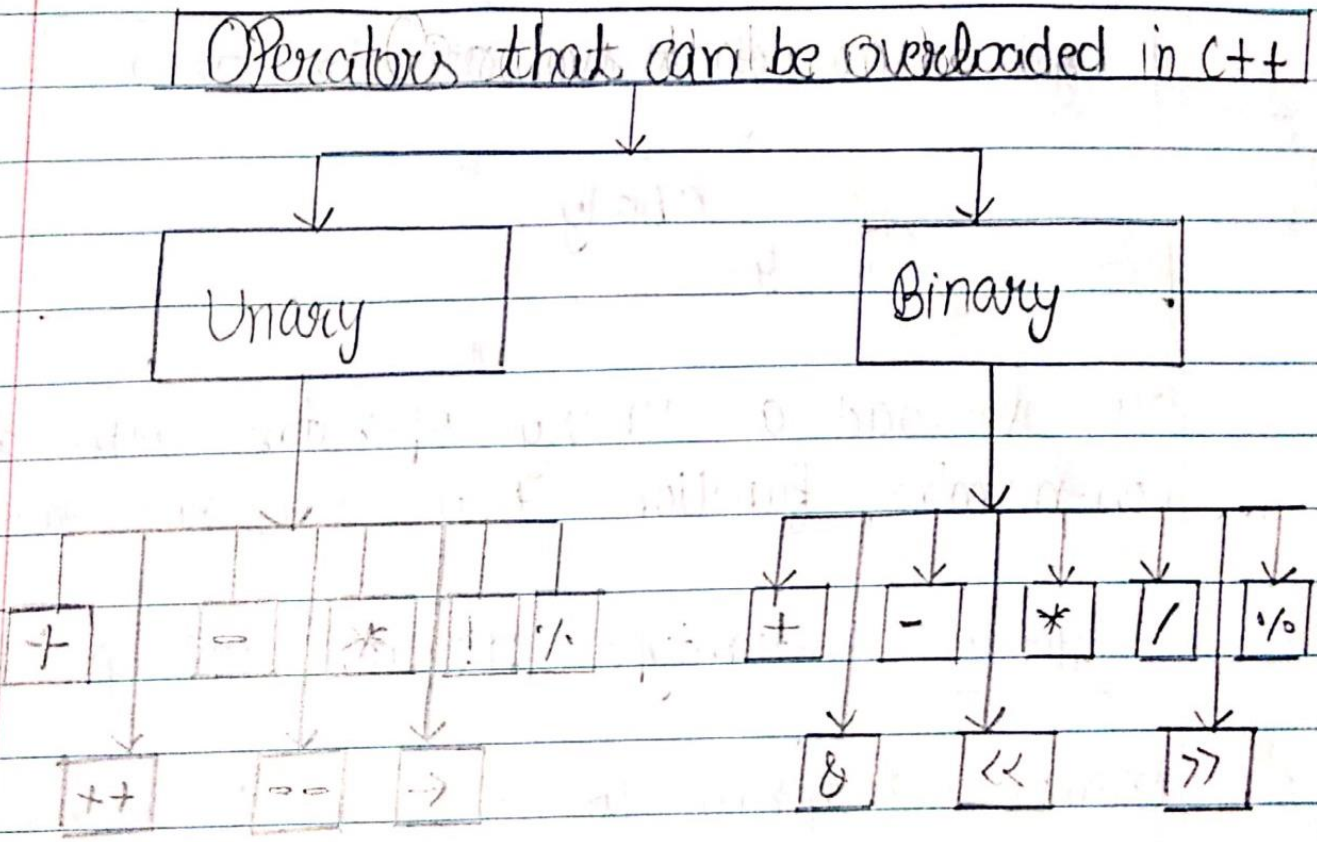
Binary Operator declared as member function take one argument; if declared as global functions, they take two arguments.

The operator symbol is used after using the operator reserve word.

Types of Operator Overloading:-

These are categorized into two parts as:-

(i) Unary Operator Overloading
(ii) Binary Operator Overloading



(i) Unary Operator Overloading

A operator which contain only one operand is called Unary Operator.

It is a overloading of an operator operating on a single operand.

The increment operator ++ and decrement operator -- are examples of unary operators.

We can overload a unary operator either class function that has no parameter.

Syntax:-
```
returntype operator op()
{
    //body;
}
```

Or, overload a unary operator with a non-member function that has one parameter.

Syntax: returntype operator op(arg);

Example: Program to overload Unary ++ Operator

```
#include<iostream.h>
#include<conio.h>

class abc
{
public:
```

```
int a;
void input()
 {
  a=5;
 }
void operator++()
 {
  a++;
 }
void show()
 {
  cout<<a;
 }
};

void main()
 {
  clrscr();
  abc ob;
  ob.input();
  ob++;
  ob.show();
  getch();
}
```

(ii) Binary Operator Overloading

A operator which contain two operands is called Binary Operator.

It is overloading of an operator operating on two operands.

```
result = num + 10;
```

→ + is an binary operator.
→ that works on operands num and 10.

In binary operator overloading function, there should be one argument to be passed.

Binary operators are additional to unary operator.

Syntax:-

```
datatype operator op(arg);
{
  // body;
}
```

```
#include<iostream.h>
#include<conio.h>
class abc
 {
 int a,b;
 public
 abc(int p, int q)
 {
  a=p;
  b=q;
 }
 void show()
 {
cout<<"a is"<<a<<"b is"<<b;
}
abc operator +(abc obj)
 {
 abc temp(0,0);
 temp.a = a+obj.a;
 temp.b = b+obj.b;
 return temp;
 }
};

void main()
 {
```

```
clrscr();
abc ob(10,20), obj1(20,40), obj2(0,0);
obj2 = ob + obj1;
obj2.show();
getch();
}
```

Output:- a is 30 b is 62

(Difference)

	Overloading Unary Operator	Overloading Binary operator.
1	Operator overloading function works on single value.	1 Overloading Binary operator works on two values.
2.	No of objects passed to the function is one only.	2) No of objects passed to the functions are two.
3	It is simple to use.	3) It is little difficult.
4.	Example: (-) minus operator	4) Example: (<) less than