Object oriented Programming using C++ 4th Sem
unit 5th

Operator Overloading- operator overloading is one of the many exciting features of C++ Language.

→ operator overloading provides a flexible option for the creation of new defination for most of the C++ operators.

We can overload all the C++ operators except the following.

→ Class member access operators (.,*)

→ scope Resolution operator (::)

→ size operator (sizeof)

→ conditional operator (?:)

- We can achieve polymorphism on the basis of operator overloading in C++. (it is a type of Polymorphism)
- → we can perform many actions with a single operator.
- → To define an additional task to an operator, we must specify
- → (operator) overloaded operator is used to perform operation on user defined data type.

operator overloading Syntax -

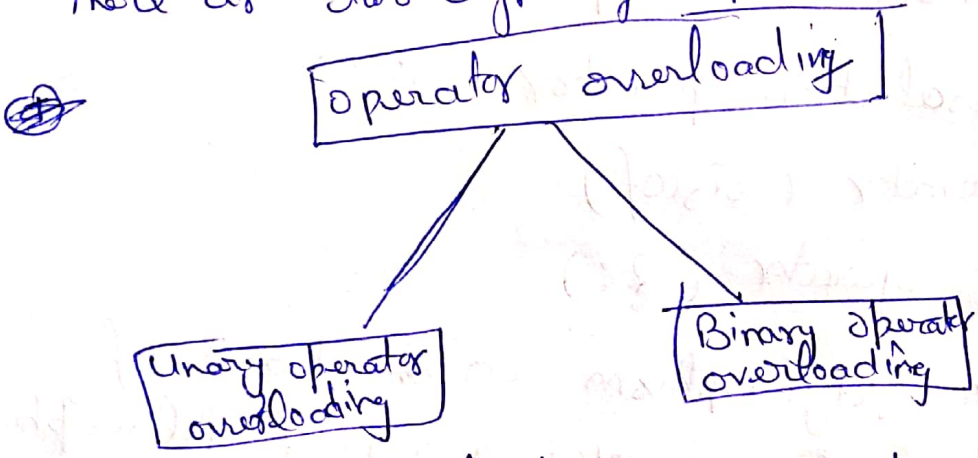
keyword ↑ operator to be overloaded ↑

```
ReturnType classname :: operator OperatorSymbol(arg lis
{
    // function Body }
```

operator overloading can be done by implementing a function which can be:

1. Member function
2. Non " "
3. Friend function.

There ar two types of operator overloading



unary operator overloading - unary operator are those operator which works on single operand.

→ unary operators (+), (−), (++), (−−) and !.

→ unary function are overloaded a member function without any formal arguments.

WAP in C++ to overload a unary minus (−) operator.

Program -

```
#include <iostream.h>
Class First
{
    int x, y;
  Public:
    void getdata (int a, int b);
    void display ();
    void operator - ();  // Declare overload unary minus
};
void First :: getdata (int a, int b)
{
    x = a; y = b;
}
void First :: display (void)
{
    cout << "x =" << x;
    cout << "y = " << y;
}
void First :: operator - ()  // Defining unary - operator
{
    x = -x;
    y = -y;
}
void main ()
{
    First f;
    f.getdata (17, -34)
    -f;  // Calling operator - () function
    cout << "After Calling unary operator f =";
    f.display ();
    return 0;
}
```

o/p is =
-17, 34

overloading binary operators - Binary operator work on two perands.

→ Binary operator can be overloaded with the help of member function after passing formal arguments

→ Arithmatic operators +, -, *, /, % overloaded as binary op^r.

ex WAP. in C++ to overload a binary operator to add complex Numbers.

```
# include <iostream>
  using namespace std;
class complex {
        float x;
        float y;
    public:
        complex() { }                                   //Cons 1
        complex(float real, float imag)                 //Cons 2
        { x = real;  y = imag; }
        complex operator + (complex);
        void display (void);
};
complex complex :: operator + (complex c)
{ complex temp;
  temp.x = x + c.x;
  temp.y = y + c.y;
  return (temp); }
```

Contd.

```
void complex :: display (void)
  {
  cout << x << " +j " << y << "\n";
    }
int main()
  {
  complex c1, c2, c3;
  c1 = complex (2.5, 3.5);
  c2 = complex(1.6, 2.7);
  c3 = c1 + c2;
cout << "c1 = " : c1.display();
cout << " c2 = " : c2.display();
cout << "c3 = " : c3.display();
  return 0;
    }
```

output -

c1 = 2.5 + j3.5

c2 = 1.6 + j 2.7

c3 = 4.1 + j 6.2