Operator overloading

Polymorphism is a concept in object-oriented programming that allows objects of different classes to be treated as objects of a common superclass. This means that a method can be written to accept objects of a certain superclass, and it will work with objects of any subclass of that superclass.

There are two main types of polymorphism: *compile-time polymorphism,* also known as method overloading, and *runtime polymorphism*, also known as method overriding.

Operator overloading allows us to redefine the behaviour of operators such as +, -, *, / for user-defined types in programming languages.

Syntax:

```
return_type operator op (parameters) {
  // code to define the behavior of the operator
}
```

Different approaches to Operator Overloading

You can perform operator overloading by implementing any of the following types of functions:

- Member Function
- Non-Member Function
- Friend Function

The operator overloading function may be a member function when a Left operand is a object of the Class.

When the Left operand is different, the Operator overloading function should non-member function.

You can make the operator overloading function a friend function it needs

Restrictions to Operator Overloading

- The operators = (scope resolution). (member access). (member access through pointer to member), and? (ternary conditional) cannot be overloaded.
- It is not possible to change the precedence, grouping, or number of operands operators
- The overload of operator must either return a raw pointer, or return an object (by reference or by value) for which operator is in turn overloaded
- The overloads of operators & and lose stron-Circuit Evaluation