# C++ overloading (operator and function)

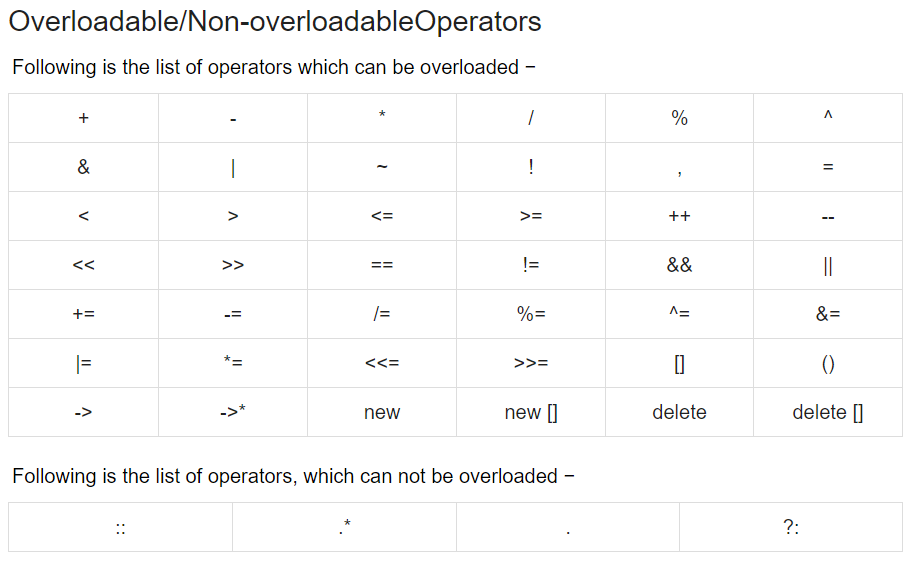
* C++ allows you to specify more than one definition for a function name or an operator in the same scope, which is called function overloading and operator overloading respectively.
* The process of selecting the most appropriate overloaded function or operator is called **overload resolution**.

## Function Overloading in C++

* Two or more functions have the same name but different number of parameters or different type of parameters are called function overloading.
* Cannot overload function declarations that differ only by return type.

## Operator Overloading in C++

* Overloaded operators are functions with special names: the keyword "operator" followed by the symbol for the operator being defined.
* Like any other function, an overloaded operator has a return type and a parameter list.



## Unary operator overloading

* The unary operator operates on 1 single operand
* Some example of unary operators: increment(++), decrement(--), minus(-), NOT(!)
* Usually these operators appear on the left side of the operand, but sometime they can be used as postfix as well.
* For post-increment and post-decrement, there must be ONE argument, which is the type of the variable that you use in the overloading part (int / double / etc.), passed to the unary operator overloading function.
* For pre-increment and pre-decrement, there must be NO argument passed to the function.

## Binary operator overloading

* The binary operators take two arguments.
* Some example of binary operators: addition(+), subtraction(-), division(/).

## Relational operator overloading

* There are various relational operators supported by C++ language like (<, >, <=, >=, ==, etc.)
* You can overload any of these operators, which can be used to compare the **objects of a class**.
* The type returned by the relational operator is **bool**.

## Input/output operator overloading

* C++ is able to input and output the built-in data types using the stream extraction operator >> and the stream insertion operator <<.
* Here, it is important to make operator overloading function a friend of the class because it would be called without creating an object.

## Increment/Decrement operator overloading

* The increment (++) and decrement (--) operators are two important unary operators available in C++.

## Assignment operator overloading

* You can overload the assignment operator (=) just as you can other operators and it can be used to create an object just like the copy constructor.
* Assignment operator return void type.

## Function call ( ) operator overloading

* The function call operator () can be overloaded for objects of class type.
* When you overload ( ), you are not creating a new way to call a function. Rather, you are creating an operator function that can be passed an arbitrary number of parameters.

## Subscripting [] operator overloading

* The subscript operator [] is normally used to access array elements. This operator can be overloaded to enhance the existing functionality of C++ arrays.