Rules for Overloading Operators

Rules for Overloading Operators:

- Only Existing operators can be overloaded.
- The overloaded operator must have at least one operand is of user defined type.
- We cannot change the basic meaning of an operator. That is to say, we cannot redefine the plus (+) operator to subtract one value from the other.
- Overloaded operators follow the syntax rules of the original operators. They cannot be overridden.

Rules for Overloading Operators:

➢There are some operatos that cannot be overloaded. These are given below:

❑ Class member access operators	**.**
❑Class member access operators	.*
❑Scope resolution operator	**::**
❑Size Operator	**sizeof**
❑Conditional Operator	?:

Rules for Overloading Operators:

- We cannot use friend functions to overload certain operators. The list is given below:

❑Assignment Operator	=
❑Function call Operator	( )
❑Subscription Operator	[ ]
❑Class member access operator	->

- However member function can be used to overload the above operators.

Rules for Overloading Operators:

- Unary operators, overloaded by means of a member function, take no explicit arguments and return no explicit values, but, those overloaded by means of a friend function, take one reference argument (the object of the relevant class).
- Binary operator overloaded through a member function take one explicit argument and those which are overloaded through a friend function take two explicit argument.
- When using binary operators overloaded through a member function, the left hand operand must be an object of relevant class.

Rules for Overloading Operators:

- Binary arithmetic operators such as +, -, * and / must explicitly return a value. They must not attempt to change their own arguments.