1. Except for the overloaded function-call operator, **operator()**, an overloaded operator may not have default arguments.
2. When an overloaded operator is a member function, **this** is bound to the left-hand operand. Member operator functions have one less (explicit) parameter than the number of operands.
3. Ordinarily, the comma, address-of, logical AND, and logical OR operators should not be overloaded.
4. Symmetric operators—those that might convert either operand, such as the arithmetic, equality, relational, and bitwise operators—usually should be defined as ordinary nonmember functions.
5. Generally, **output operators** should print the content of the object, with minimal formatting. They should not print a newline.
6. **Input operators** must deal with the possibility that the input might fail; output operators generally don’t bother.
7. Input operators should decide what, if anything, to do about error recovery.
8. If a single logical definition for **<** exists, classes usually should define the **< operator**. However, if the class also has **==,** define **<** only if the definitions of **<** and **==** yield consistent results.
9. Assignment operators can be overloaded. Assignment operators, regardless of parameter type, must be defined as member functions.
10. If a class has a **subscript operator**, it usually should define two versions: one that returns a plain **reference** and the other that is const member and returns a **reference to const**.
11. The overloaded **arrow operator** ***must*** (unlike other overloaded operators) return either a pointer to a class type or an object of a class type that defines its own operator arrow.
12. A call signature corresponds to a function type.
13. different types can have the same call signature. [Example](Overloaded%20Operations%20and%20Conversions.cpp)