|  |  |
| --- | --- |
| 1. 基本概念 2. 直接调用一个重载的运算符函数 3. 某些运算符不应该被重载 4. 使用与内置类型一致的含义 5. 赋值和复合赋值运算符 6. 选择为成员或者非成员 | Cout<<item1+item2  Data1+data2;  operator+(data1,data2);//非成员函数  data1.operator(data2);//成员函数  string t=s+”!”;s需定义运算符+,第一个参数是string  string u=”hi”+s;//必须非成员，第二个参数是stirng才可 |
| 1. 基本概念 2. 重载运算符是具有特殊名字的函数：由operator和其后定义的运算符号共同组成，同样有返回类型，参数列表和函数体 3. 参数与该运算符作用的运算对象一样多 4. 除了重载的函数调用运算符operator()之外，其他重载运算符不能含有默认实参 5. 如果运算符函数是成员函数，第一个（左侧）运算对象绑定到隐式的this指针上，所以参数数量比运算符的运算对象总数少一个 6. 或是类的成员，或者至少含有一个类类型的参数。当运算符作用于内置类型的运算对象时，我们无法改变该运算符的含义   就是不能为int重定义内置的运算符  Int operator(int,int);   1. 只能重装已有的运算符，不能新的运算符号 2. 重载的运算符，其优先级和结合律与对应的内置运算符保持一致 3. 直接调用一个重载的运算符函数   非成员运算符函数   1. 可以调用普通函数一样直接调用运算符函数,operator+(data1,data2) 2. 运算符作用于类型正确的实参 （间接）就是，data1+data2   成员函数   1. 用点 或者 箭头 2. 或间接，datat1+=data2; 3. 某些运算符不应该被重载 4. 因为指定了运算对象求值顺序，无法应用到重载的运算符上 5. 不应该重载逗号、取地址、逻辑与和逻辑或 6. 使用与内置类型一致的含义 7. 重载运算符的返回类型通常情况下应该与其内置版本的返回类型兼容,不要扭曲运算符常规 含义 8. 赋值和复合赋值运算符 9. +=应该执行重载的+运算符 10. 选择为成员或者非成员 11. 赋值、下标、调用 箭头运算符必须是成员函数 12. 复合赋值一般是成员，不是必须 13. 改变对象状态的运算符或者与给定类型密切相关的运算符，递增、递减、解引用 通常是成员 14. 对称性，可能转换任意一端的运算对象，算术、相等性、关系、位运算符等，通常非成员函数 15. 运算符定义成成员函数时，左侧运算对象必须是运算符所属类的一个对象 16. 定义成非成员函数时，每个实参都能被转换成形参类型，唯一要求至少需要一个运算对象是类类型   代码：  string s = "world";  string s2 = s + "!";  string s3 = "hi" + s2;// 如果+是string的成员，则产生错误  第二个+运算符意思是，除非定义stirng的非成员重载+运算符，第一个运算符对象是const char\*，第二个运算符是string，才不会产生错误 | |

