重载运算符

格式：operator*op*(); 例如operator+()表示重载+运算。

示例：

// 声明

class Time{

…

public:

Time operator+(const Time &t) const;

…};

// 定义

Time Time::operator+(const Time &t) const{

Time sum;

sum.minutes = minutes+t.minutes;

sum.hours = hours+t.hours+sum.minutes/60;

sum.minuts %/60;

return sum;}

// 调用

Time total, c, f;

total = c.operater+(f);//方法1

total = c+f;//方法2

符号重载的限制和可重载的运算符参见P387

友元。通过让函数成为类的友元，可以赋予函数与类的成员函数相同的访问权限。

友元函数属于非成员函数。创建友元函数第一步是将其原型函数放入类声明中：

friend Time operator\*(double m, const Time &t);//声明

Time operator\*(double m, const Time &t){…}//定义

示例，重载<<运算符。P392

对于很多运算符来说，可以使用成员函数或非成员函数两种方法实现重载，一般来说，非成员函数是友元函数。如加法运算符需要两个操作数，成员函数版本中，一个操作数通过this指针隐式地传递，另一个操作数作为函数参数显式地传递；对于友元版本，两个操作数都是作为参数来传递。

强制类型转换