CSDN 学院

C++语言基础

迁者 - 贺利坚

http://blog.csdn.net/sxhelijian/

http://edu.csdn.net

本节主题：

不同类型数据间的转换

不同类型数据间的转换

- 在C++中，某些标准类型的数据之间可以自动转换
- 隐式类型转换：由C++编译系统自动完成的，用户不需干预
  - int i = 6;
  - i = 7.5 + i;
- 强制(显式)类型转换：在程序中将一种类型数据明确转换成另一指定的类型
  - int(89.5) , float(56)
- 问题：
  - 用户自己声明的类型，编译系统并不知道怎样进行转换(例Complex<==>double)
- 解决办法：
  - 定义专门的函数来，让编译系统知道怎样去进行这些转换。
  - 转换构造函数和类型转换函数

类的几种构造函数：可以同时出现在同一类中(构造函数的重载)

- 默认构造函数
  - Complex( );　　//没有参数
- 用于初始化的构造函数
  - Complex(double r,double i); //常有两个以上的形参
- 用于复制对象的复制构造函数
  - Complex (Complex &c); //本类对象的引用作形参
- 转换构造函数(conversion constructor function) ——*new!*
  - 将一个其他类型的数据转换成一个类的对象的构造函数
  - Complex::Complex(double r) {real=r;imag=0;} //完成指定类型到本类对象的转换

转换构造函数及应用

- Complex::Complex(double r) {real=r;imag=0;}
- 应用举例

```
Complex c1(3.5);
Complex c2,c3;
c2=Complex(3.6);   //执行转换构造函数
c3=c1+2.5;   //非法，除非定义了相应的运算符重载函数
c3=c1+Complex(2.5);  //合法，利用转换构造函数，相当于强制类型转换
```

- 通常把有一个参数的构造函数用作类型转换，而不作别的用途。反例：

```
Complex::Complex(double r) {cout<<r;}
Complex::Complex(double r) {real=0;imag=r;}
```

类型转换函数

Complex(double r) {real=r;imag=0;}
//double==>Complex

- 问题
  - 用转换构造函数可以将一个标准类型的数据转换为类的对象；
  - 转换构造函数不能反过来将一个类的对象转换为一个标准类型的数据

Complex==>double ???

- 解决
  - 用类型转换函数(type conversion function)将一个类的对象转换成另一类型的数据
- 例
  - 如果已声明了一个Complex类，可以在Complex类中定义类型转换函数（成员函数）
  - Complex::operator double( ) {return real;}

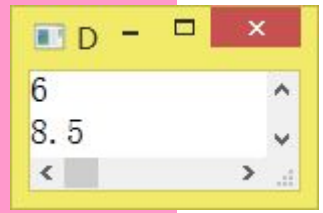
类型转换函数应用

```
class Complex
{
public:
    Complex( ){real=0; imag=0;}
    Complex(double r,double i){real=r; imag=i;}
    operator double( );
    friend Complex operator + (Complex c1,Complex c2);
private:
    double real;
    double imag;
};
```

```
Complex operator + (Complex c1,Complex c2)
{
   return Complex(c1.real+c2.real, c1.imag+c2.imag);
}
```

```
Complex::operator double( )
{
   return real;
}
```

```
int main( )
{
    Complex c1(3.5,4),c2(5,-10);
    double d1,d2;
    d1=2.5+c1;
    cout<<d1<<endl;
    d2=c1+c2;
    cout<<d2<<endl;
    return 0;
}
```



应用：当需要的时候，编译系统会根据表达式的上下文，自动调用类型转换函数，将Complex类对象作为double类型数据使用

关于类型转换函数的应用

- 类型转换函数也称：类型转换运算符函数、类型转换运算符重载函数、强制类型转换运算符重载函数
- 不同类型进行各种混合运算的方案
  - 转换构造函数
  - 类型转换函数
  - 运算符重载
- 进行各种运算时，使用类型转换函数，而不是对多种运算符进行重载，工作量较小，程序精干
  - 例：对一个Complex类对象和一个double型变量进行+,-, *,/等算术运算，以及关系运算和逻辑运算
- 防止出现二义性

```
int main( )
{
   Complex c1(3.5,4),c2(5,-10);
   double d1,d2;
   d1=2.5+c1;
   cout<<d1<<endl;
   d2=c1+c2;
   cout<<d2<<endl;
   return 0;
}
```

CSDN 学院

THANKS

本课程由 迁者-贺利坚 提供

CSDN网站：www.csdn.net
企业服务：http://ems.csdn.net/
人才服务：http://job.csdn.net/
CTO俱乐部：http://cto.csdn.net/
高校俱乐部：http://student.csdn.net/
程序员杂志：http://programmer.csdn.net/

CODE平台：https://code.csdn.net/
项目外包：http://www.csto.com/
CSDN博客：http://blog.csdn.net/
CSDN论坛：http://bbs.csdn.net/
CSDN下载：http://download.csdn.net/