西北工业大学
NORTHWESTERN POLYTECHNICAL UNIVERSITY

# C++程序设计
# Programming in C++

**1011018**

主讲：魏英，计算机学院

# 默认构造函数和复制构造函数

1、定义默认构造函数

2、隐式类类型转换

- 默认构造函数就是在没有显式提供初始化式时调用的构造函数，它是一个不带参数的构造函数。如果定义某个类的对象时没有提供初始化式就会使用默认构造函数。

- 定义默认构造函数（default constructor）的一般形式为：

```
类名()
{
        函数体
}
```

- 它由不带参数的构造函数，或者所有形参均是默认参数的构造函数定义。与默认构造函数相对应的对象定义形式为：

```
类名 对象名;
```

- 任何一个类有且只有一个默认构造函数。如果定义的类中没有显式定义任何构造函数，编译器会自动为该类生成默认构造函数，称为合成默认构造函数（synthesized default constructor）。
- 一个类哪怕只定义了一个构造函数，编译器也不会再生成默认构造函数。换言之，如果为类定义了一个带参数的构造函数，还想要无参数的构造函数，就必须自己定义它。
- 一般地，任何一个类都应定义一个默认构造函数。因为，在很多情况下，默认构造函数是由编译器隐式调用的。

- 为了实现其他类型到类类型的隐式转换，需要定义合适的构造函数。可以用单个实参调用的构造函数（称为转换构造函数）定义从形参类型到该类类型的隐式转换。

## 【例28.1】隐式类类型转换举例

```
#include <iostream>
using namespace std;
class Data { //Data类定义
public:
  Data(const string& str="") : s1(str) { }
  void SetString(const Data& r) //期待的是Data类型的对象
  { s1=r.s1; }
  void print() { cout<<s1<<endl; }
private:
  string s1;
};
```

```
13 int main()
14 {
15     Data a,b,c("world");
16     string i="string";
17     a.SetString(c);
18     b.SetString(string("world")); //隐式转换
19     a.print();   b.print();
20     Data d=Data(i);  //隐式转换
21     d.print();
22     return 0;
23 }
```

- 使用单个参数的构造函数来进行类类型转换的方法可以总结如下：
- （1）先声明一个类；
- （2）在这个类中定义一个只有一个参数的构造函数，参数的类型是需要转换的数据类型，即转换构造函数的一般形式为：

```
类名(const 指定数据类型& obj)
```

- （3）采用转换构造函数定义对象时即进行类型转换，一般形式为：

```
类名(指定数据类型的数据对象)
```

- 可以禁止由构造函数定义的隐式转换，方法是通过将构造函数声明为explicit，来防止在需要隐式转换的上下文中使用构造函数。

- C++关键字explicit用来修饰类的构造函数，指明该构造函数是显式的。explicit关键字只能用于类内部的构造函数声明上，在类定义外部不能重复它。

- 一般地，除非有明显的理由想要定义隐式转换，否则，单形参构造函数应该为explicit。将构造函数设置为explicit可以避免错误，如果真需要转换，可以显式地构造对象。

# C程序设计