

C++语言基础

迂者 - 贺利坚

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

<http://edu.csdn.net>





本节主题：

函数重载

重载函数：同名同体，但接口不同

```
#include <iostream>
using namespace std;
int max(int a,int b,int c);    //函数声明
double max(double a,double b,double c);
long max(long a,long b,long c);
int main( )
{
    int i1,i2,i3,i;
    cin>>i1>>i2>>i3;    //输入3个整数
    i=max(i1,i2,i3);    //求3个整数中的最大者
    cout<<"i_max="<<i<<endl;
    double d1,d2,d3,d;
    cin>>d1>>d2>>d3;    //输入3个双精度数
    d=max(d1,d2,d3);    //求3个双精度数中的最大者
    cout<<"d_max="<<d<<endl;
    long g1,g2,g3,g;
    cin>>g1>>g2>>g3;    //输入3个长整数
    g=max(g1,g2,g3);    //求3个长整数中的最大者
    cout<<"g_max="<<g<<endl;
}
```

```
int max(int a,int b,int c)
{
    if(b>a) a=b;
    if(c>a) a=c;
    return a;
}
double max(double a,double b,double c)
{
    if(b>a) a=b;
    if(c>a) a=c;
    return a;
}
long max(long a,long b,long c)
{
    if(b>a) a=b;
    if(c>a) a=c;
    return a;
}
```

本例中，重载的函数名相同，参数、函数体相同，仅数据类型不同。

重载函数：同名不同体，参数个数有区别

```
#include <iostream>
using namespace std;
int max(int a,int b,int c); //函数声明
int max(int a,int b);
int main( )
{
    int a=8,b=-12,c=27;
    //输出3个整数中的最大者
    cout<<"max(a,b,c)="<<max(a,b,c)<<endl;
    //输出两个整数中的最大者
    cout<<"max(a,b)="<<max(a,b)<<endl;
}
```

```
int max(int a,int b,int c)
{
    if(b>a) a=b;
    if(c>a) a=c;
    return a;
}

int max(int a,int b)
{
    if(a>b) return a;
    else return b;
}
```

重载函数除了允许参数类型不同以外，还允许参数的个数不同。

函数的重载(function overloading): 一名多用

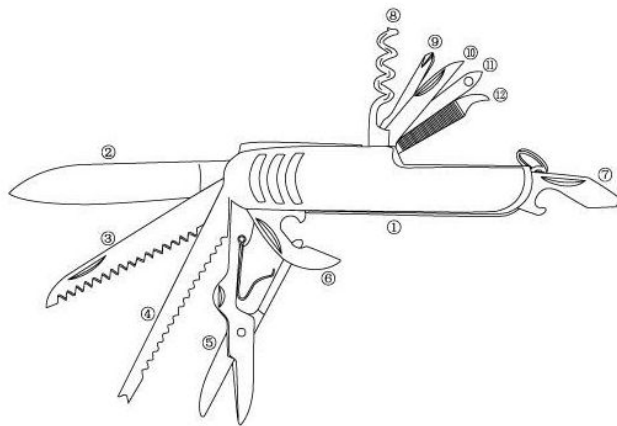
- ❏ 函数的重载：用同一函数名定义多个函数，这些函数的参数个数或参数类型不同。
- ❏ 函数的重载使一个函数名重新赋予新的含义，使一个函数名可以多用。
- ❏ 在编程时，要实现细节不同的同一类的功能时，用函数的重载。例如

```
int max(int a , int b, int c);
```

```
double max(double a , double b , double c);
```

```
long max(long a , long b , long c);
```

```
int max(int a , int b);
```



不能算做重载的情况

- 函数的返回值类型不同，但参数个数和类型相同，不是重载

```
int f(int);
```

```
long f(int);
```

```
void f(int);
```

调用f(10)?



- 重载函数的参数个数、参数类型或参数顺序三者中必须至少有一种不同，函数返回值类型可以相同也可以不同。
- 在使用重载函数时，同名函数的功能应当相同或相近，**不要用同一函数名去实现完全不相干的功能**，虽然程序也能运行，但可读性不好，使人莫名其妙。

函数重载与参数的默认值不要冲突

```
#include <iostream>
using namespace std;
int max(int a,int b,int c=5); //函数声明
int max(int a,int b);
int main( )
{
    int a=8,b=-12,c=27;
    cout<<"max(a,b,c)="<<max(a,b,c)<<endl;
    cout<<"max(a,b)="<<max(a,b)<<endl;
    return 0;
}
```

error—call of overloaded
'max(int&, int&)' is ambiguous
调用重载函数引发歧义

```
int max(int a,int b,int c)
{
    if(b>a) a=b;
    if(c>a) a=c;
    return a;
}

int max(int a,int b)
{
    if(a>b) return a;
    else return b;
}
```

函数的重载的真正威力

❏ 自定义类型中，运算符赋予新含义

```
Complex operator+(Complex& c1,Complex& c2)
```

```
istream& operator>>(istream& output, Complex& c)
```

```
Complex c1, c2,c3;
```

```
cin>>c1>>c2;
```

```
c3=c1+c1;
```

```
Time t1(12,25,34),t2(4,38,45);
```

```
int delay=t1-t2;
```

```
cout<<(t1+180)<<end;
```


THANKS

本课程由 迂者-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/>

