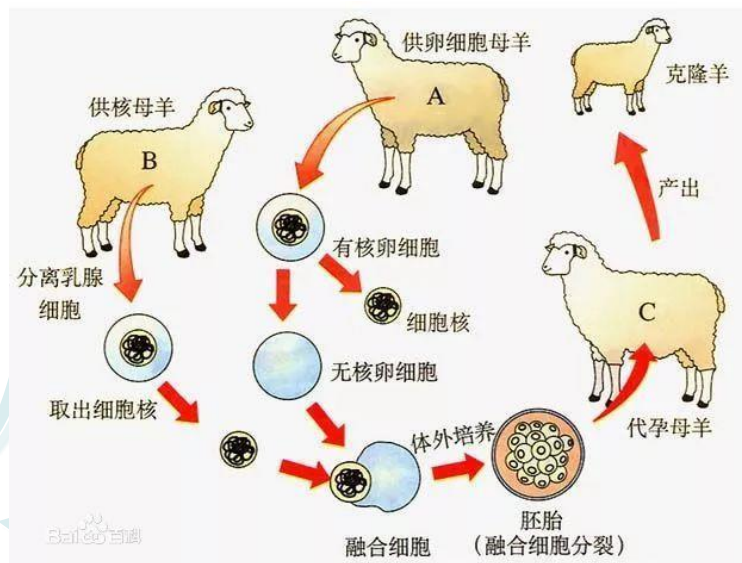


# C/C++程序设计案例实战

华中农业大学信息学院 章 英

# 问题引入



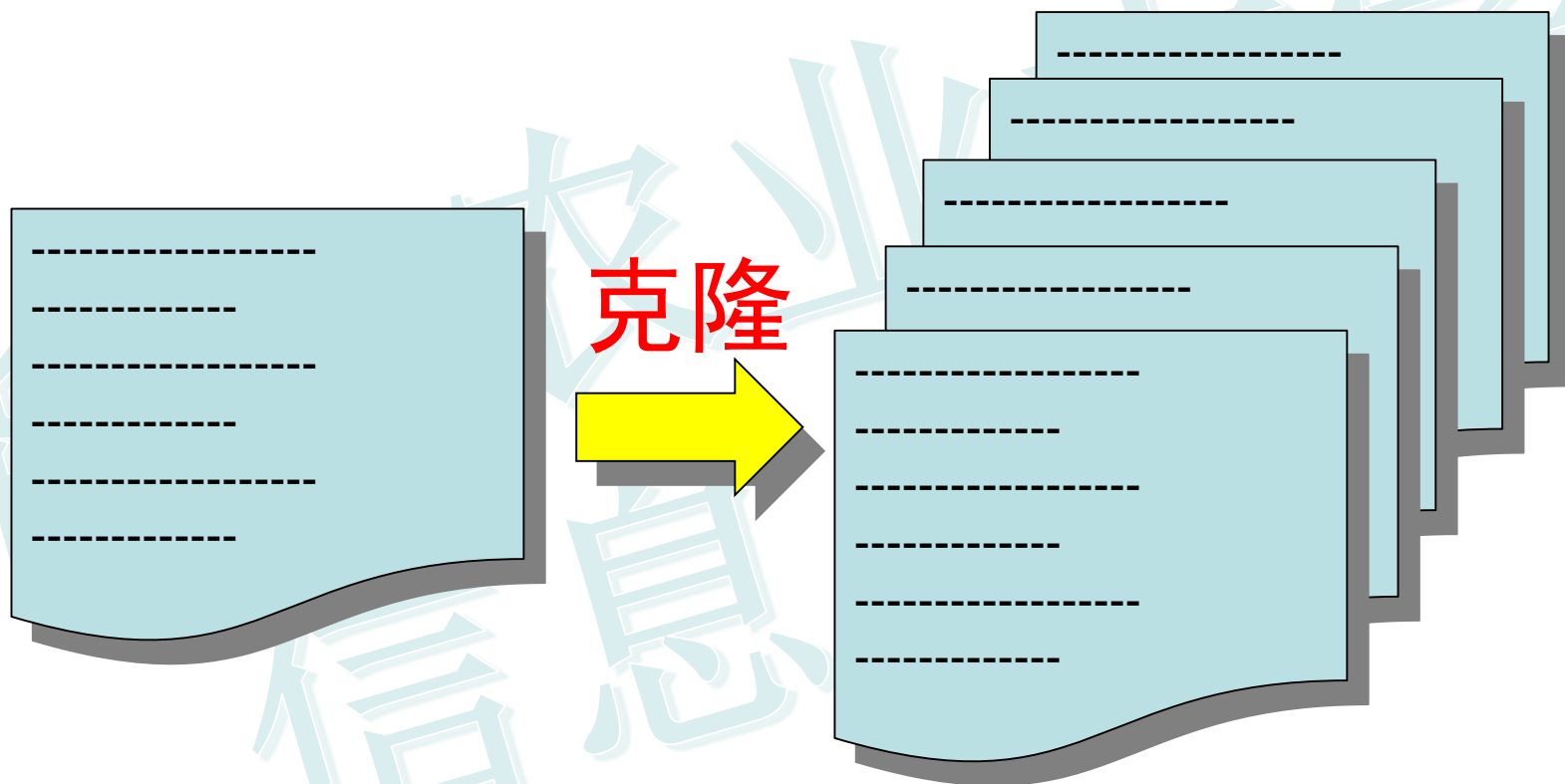
## 克隆

刻苦学习  
努力钻研  
创造奇迹



## 复制、拷贝

# 案例分析



代码A

代码BCDEF...

## 模板

模板是泛型编程的基础，泛型编程即以**一种独立于任何特定类型**的方式编写代码。

## 代码重用

代码具有**通用性**，不受数据类型的影响。

# 案例分析

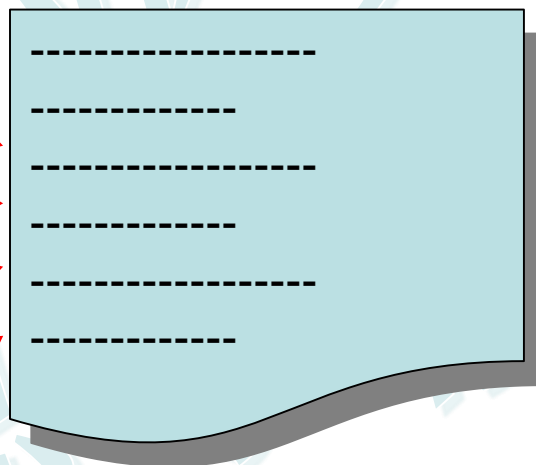
**功能：**对数据采用选择排序法  
从小到大依次排好

整形

实数型

字符串型

学生类类型



代码A

# 排序函数

```
void Sort(int a[],int n)
{
    //.....
}
```

```
void Sort(double d[],int n)
{
    //.....
}
```

```
void Sort(char s[][20],int n)
{
    //.....
}
```

```
void Sort(Student stu[],int n)
{
    //.....
}
```

## 想一想

```
void Sort(int a[],int n)
{
    //.....
}
```

```
void Sort(double d[],int n)
{
    //.....
}
```

```
void Sort(char s[][20],int n)
{
    //.....
}
```

```
void Sort(Student stu[],int n)
{
    //.....
}
```



函数  
重载

## 函数重载

- (1) 函数的参数个数不同
- (2) 函数的参数类型不同或者参数类型顺序不同

— “物” 多用



# 函数重载举例

```
void F();  
void F(int);  
void F(double, double=5.8);  
void F(const char *, const char *);
```

```
int main()  
{  
    F(1.2);  
    return 0;  
}
```

# 案例分析

```
void Sort(int a[],int n)
{
    //.....
}
```

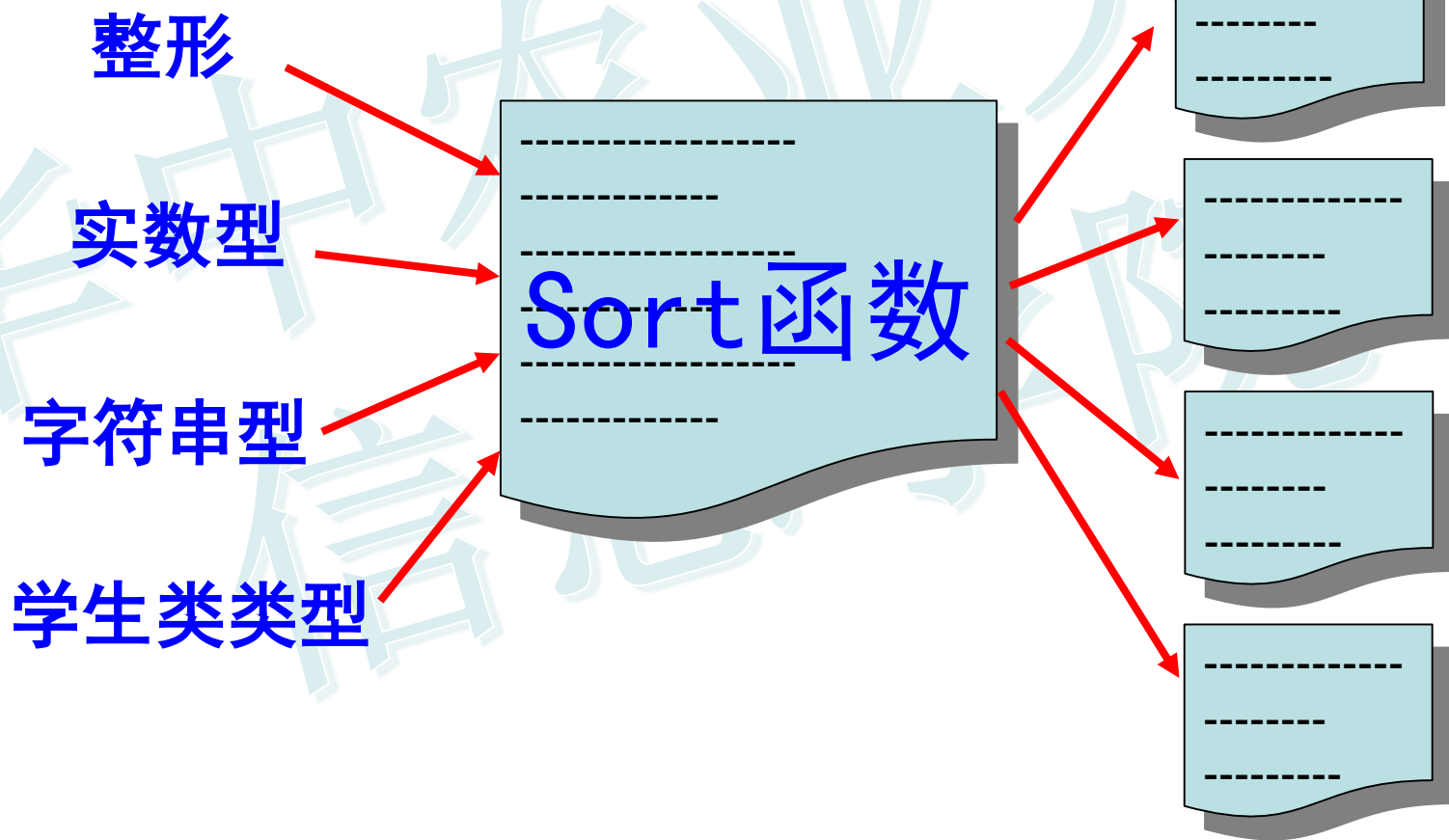
```
void Sort(double d[],int n)
{
    //.....
}
```

```
void Sort(char s[][20],int n)
{
    //.....
}
```

```
void Sort(Student stu[],int n)
{
    //.....
}
```

# 函数模板

**功能：**对数据采用选择排序法  
从小到大依次排好



# 案例——代码实现

模板关键字

模板参数列表

template <typename T>

```
27  template<typename T>
28  void Sort(T a[], int n)
29  {
30      int i,j,p;
31      T t;
32      for(i=0;i<n-1;i++)
33      {
34          p=i;
35          for(j=i+1;j<=n-1;j++)
36          {
37              if(a[p]>a[j]) p=j;
38          }
39          if(p!=i)
40              { t=a[i];a[i]=a[p];a[p]=t; }
41      }
42  }
```

# 函数模板

```
template <typename T>  
void Sort(T a[], int n)
```

int

void Sort(int a[], int n)

double

void Sort(double a[], int n)

Student

void Sort(Student a[], int n)

编译生成三个模板函数

## 案例——代码实现

`void Sort(int a[], int n)`

`void Sort(double a[], int n)`

```
27     template<typename T>
28     void Sort(T a[], int n)
29     {
30         int i,j,p;
31         T t;
32         for(i=0;i<n-1;i++)
33         {
34             p=i;
35             for(j=i+1;j<=n-1;j++)
36             {
37                 if(a[p]>a[j])    p=j;
38             }
39             if(p!=i)
40             { t=a[i];a[i]=a[p];a[p]=t; }
41         }
42     }
```

# 想一想

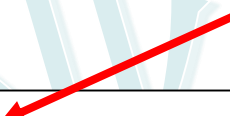
void Sort(**Student** a[], int n)

```
27  template<typename T>
28  void Sort(T a[], int n)
29  {
30      int i,j,p;
31      T t;
32      for(i=0;i<n-1;i++)
33      {
34          p=i;
35          for(j=i+1;j<=n-1;j++)
36          {
37              if(a[p]>a[j]) p=j;
38          }
39          if(p!=i)
40          { t=a[i];a[i]=a[p];a[p]=t; }
41      }
42  }
```

## 想一想

void Print(**Student** a[], int n)

关键字class



```
44     template<class T>
45     void Print(T a[],int n)
46     {
47         int i;
48         for(i=0;i<n;i++)
49             cout<<a[i]<<" ";
50         cout<<endl;
51     }
```

重载<<插入运算符



# 案例——代码实现

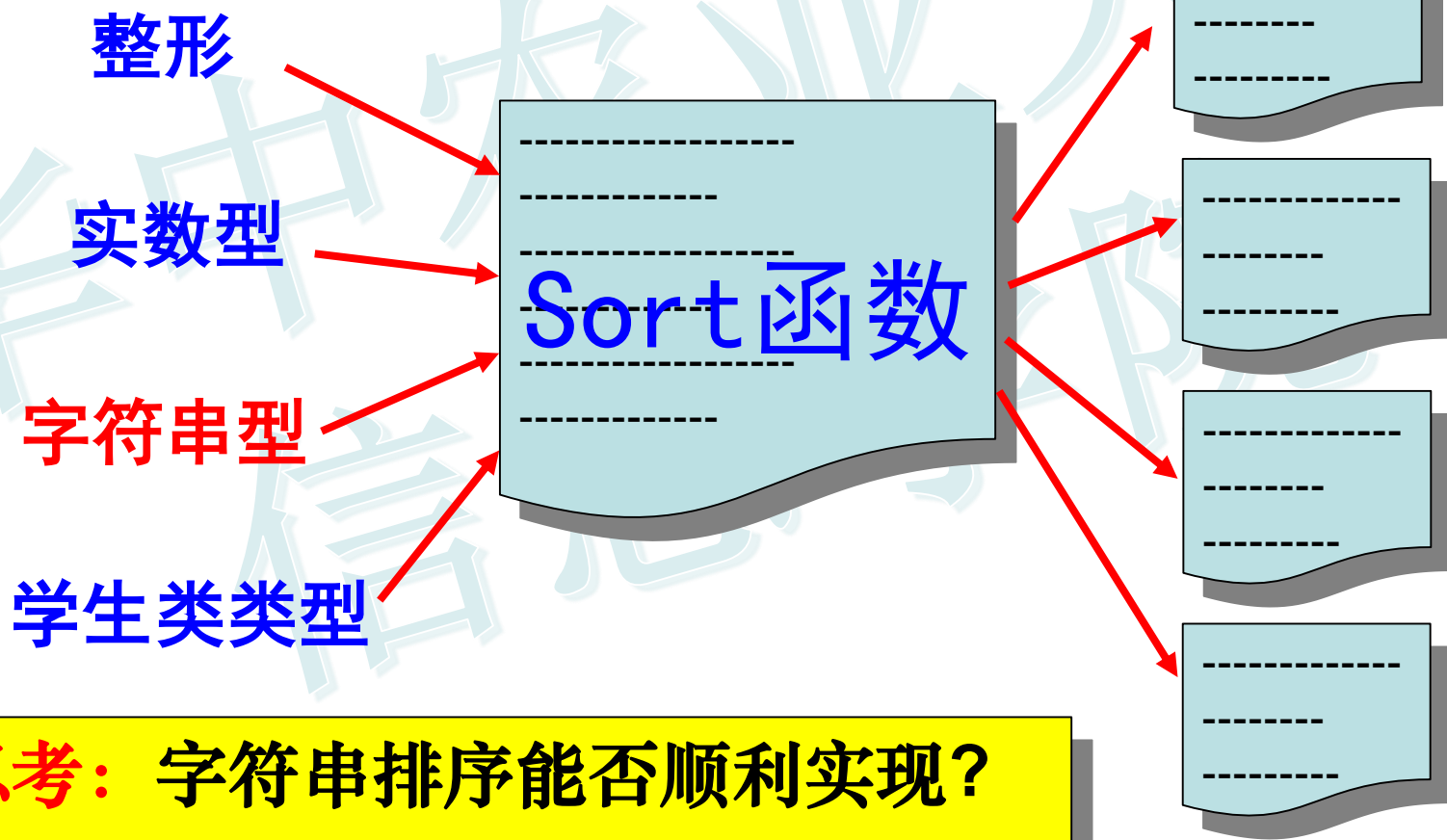
```
10.1 不同类型数据的选择排序及输出.cpp - Code::Blocks 17.12
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins Docs
Start here x 10.1 不同类型数据的选择排序及输出.cpp x
59 int main() {
60     int a[6]={4,1,2,6,3,5};
61     double d[4]={1.2,9.8,3.5,4.6};
62     Student s[3];
63     int id,score;
64     char name[20];
65     int i;
66     for(i=0;i<3;i++)
67     {
68         cin>>id>>name>>score;
69         s[i].Set(id,name,score);
70     }
71     Sort<int>(a,6);
72     Sort<double>(d,4);
73     Sort<Student>(s,3);
74     Print<int>(a,6);
75     Print<double>(d,4);
76     Print<Student>(s,3);
77     return 0;
78 }
```

```
Sort<int>(a,6);
Sort<double>(d,4);
Sort<Student>(s,3);
Print<int>(a,6);
Print<double>(d,4);
Print<Student>(s,3);
```

```
2019001 ZhangYing 90
2019002 WangFang 88
2019003 LiLei 89
1 2 3 4 5 6
1.2 3.5 4.6 9.8
2019002 WangFang 88 2019003 LiLei 89 2019001 ZhangYing 90
```

# 函数模板

**功能：**对数据采用选择排序法  
从小到大依次排好



## 小结

- (1) 能够说出构成函数重载的两个条件
- (2) 能够编程声明及实现函数模板

## 延伸

请编写一个FindMax函数模板，实现对不同类型数据查找其最大值并输出。