复旦大学

面向对象程序语言C++ 标准库

周雅倩
zhouyaqian@fudan.edu.cn
2016/3/17

复旦大学媒体计算和网络智能研究室

主要内容

- Hello World程序
- 使用字符串
- 使用批量数据
- 组织程序和数据
- 使用序列式容器
- 使用库算法
- 使用关联式容器

2016/3/17

复旦大学媒体计算和网络智能研究室

库程序的几个性质

- 解决了一个特殊的问题
- 与其它大多数库程序无关
- 有一个名字

- 组织大型程序的两种基本方法：
  – 函数
  – 数据结构

类

2016/3/17

复旦大学媒体计算和网络智能研究室

用函数组织计算

```
double grade(double midterm, double final, double homework)
{
    return 0.2 * midterm + 0.4 * final + 0.4 * homework;
}


    streamsize prec=cout.precision();
    cout << "Your final grade is " << setprecision(3)
           << 0.2 * midterm + 0.4 * final + 0.4 * median
           << setprecision(prec) << endl;
替换为:
    streamsize prec=cout.precision();
    cout << "Your final grade is " << setprecision(3)
           <<        grade(midterm,final, median)
           << setprecision(prec) << endl;
```

2016/3/17

复旦大学媒体计算和网络智能研究室

查找中值

```
typedef std::vector<double>::size_type vec_sz;
vec_sz size = homework.size();
if(size==0){
        cout << endl <<"You must enter your grades."
                << "Please try agian." <<endl;
        return 1;
}
sort(homework.begin(),homework.end());
vec_sz mid = size/2;
double median = size %2 == 0 ?
(homework[mid]+homework[mid-1])/2 :
homework[mid];
```

```
#include <stdexcept>
using std::domain_error;

double median(vector<double> vec)
{
    typedef std::vector<double>::size_type vec_sz;
    vec_sz size = vec.size();
    if(size == 0)
          throw domain_error("median of an empty vector");
    sort(vec.begin(),vec.end());
    vec_sz mid = size/2;
    return size %2 == 0 ? (vec[mid]+vec[mid-1])/2 : vec[mid];

}
```

2016/3/17

复旦大学媒体计算和网络智能研究室

异常exception

throw domain_error("median of an empty vector");

- 当一个程序抛出一个异常时，程序的执行就会停止在程序出现throw的地方，并且把异常对象传递到程序的另一个部分.
- 异常对象中含有调用程序可以用来处理异常的信息.

- domain_error: 用来说明一个函数的实参是这个函数不能接受的。

2016/3/17

重新实现计算最终成绩的方法

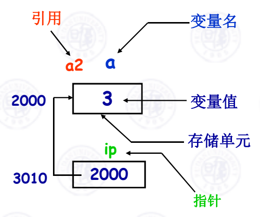
const形参　引用

```
double grade(double midterm, double final, const vector<double> & hw)
{
    if(hw.size()==0)
          throw domain_error("student has done no homework");
    return grade(midterm, final, median(hw));
}
```

函数重载

引用(reference)的基本概念

- 标识对象的方法：
  - 对象的名
  - 指向对象的指针
  - 对象的引用
- 引用是一种映射
- 引用是给标识符起别名



说明引用的一般形式

类型说明符 & 标识符(左值表达式);
类型说明符 & 标识符=左值表达式;

```
int i;
int &ii=i;
int j,&jj=j;
```

引用 Vs. 指针

- 指针是个变量，任何时候都可以向它赋值某个符合要求的变量的地址。
- 建立引用时必须进行初始化，并且不会再映射到其他变量，直至引用不再有效为止。

说明引用是一种映射的例子

```
#include <stdio.h>
void main()
{
      int a[]={0,2,4,6,8}, j;
      printf("Enter j ");
      scanf("%d",&j);
      int &ref=a[j];

      ref=44;
      printf("%d\n",ref);
      for(j=0;j<5;j++)
              printf("%d\t",a[j]);
      printf("\n");

      ref=a[0];
      printf("%d\n",ref);
      for(j=0;j<5;j++)
              printf("%d\t",a[j]);
      printf("\n");
}
```

什么能被引用，什么不能被引用

- 如果一个标识符被声明为T&的引用时，它必须用T类型的变量给这个引用初始化。
- 变量能被引用，如：指针
- 不能有void类型的引用
- 不能建立引用的数组
- 没有引用的引用，没有引用的指针
- 引用不能用类型来初始化
- 没有空的引用

引用形参

- 语言引入引用的主要目的：让函数能设置引用形参
- 函数调用时，引用形参成为实参的别名，函数体中形参的出现就是实参。
- 函数中对以形参标识的对象的引用就是对实参变量的引用，或访问实参变量，或修改实参变量。

const and 引用

- vector<double> homework;
- vector<double> & hw=homework;

- //const保证函数不会对chw做任何改变它的值的操作
- const vector<double> & chw=homework;

- vector<double> & hw1=hw;//引用的引用
- const vector<double> & chw1=chw;
- vector<double> & hw2=chw;//error!

const引用形参

- 引用形参要求系统允许函数直接访问相关的实参，而不是复制实参；
- const保证不会改变实参的值.

重载（overload）

```
double grade(double midterm, double final, const vector<double> &
    hw){
    if(hw.size()==0)
         throw domain_error("student has done no homework");
    return grade(midterm, final, median(hw));
}

double grade(double midterm, double final, double homework)
{
    return 0.2 * midterm + 0.4 * final + 0.4 * homework;
}
```

两个函数名字相同，但是参数不同

返回引用的函数

- 函数返回值时，系统要生成一个值的副本。
- 函数返回引用时，不生成引用的副本。
- 函数返回引用能使函数调用可以写在赋值表达式等号的左边
- 函数不能返回对自动变量的引用

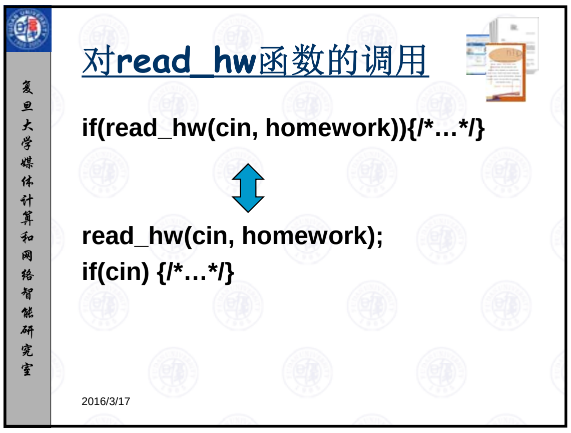
```
int & f(int a[],int index)
{
    return a[index];
}
int & g(int a[],int index)//error!
{
    int r=a[index];
    return r;
}
```

```
#include <stdio.h>
void main()
{
    int i,s[]={0,2,4,6,8};
    i=f(s,3);
    printf("%d\n",i);
    f(s,3)=22;
    printf("%d\n",f(s,3));
    for(i=0;i<5;i++)
         printf("%d\t",s[i]);
}
```

读取家庭作业成绩

```
#include <istream>
using std::istream;

istream & read_hw(istream & in, vector<double> & hw)
{
    ……//待写代码
    return in;
}
```

必须为引用形参
传递一个左值

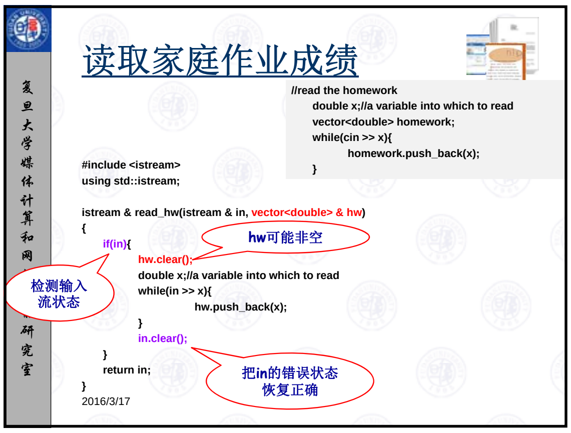


对read_hw函数的调用

```
if(read_hw(cin, homework)){/*…*/}
```

```
read_hw(cin, homework);
if(cin) {/*…*/}
```

2016/3/17



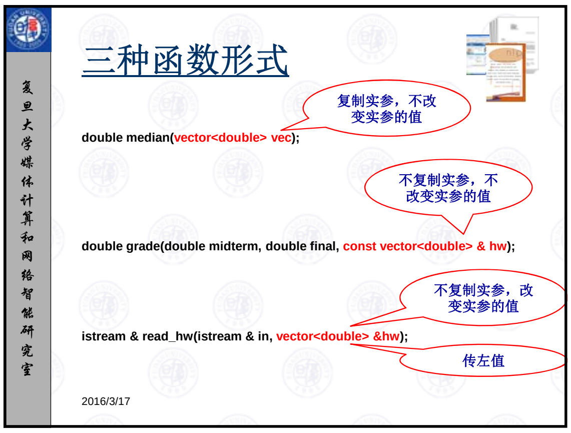
读取家庭作业成绩

```
//read the homework
    double x;//a variable into which to read
    vector<double> homework;
    while(cin >> x){
            homework.push_back(x);
    }
```

```
#include <istream>
using std::istream;

istream & read_hw(istream & in, vector<double> & hw)
{
    if(in){
            hw.clear();
            double x;//a variable into which to read
            while(in >> x){
                    hw.push_back(x);
            }
            in.clear();
    }
    return in;
}
```

hw可能非空
检测输入流状态
把in的错误状态恢复正确

2016/3/17



三种函数形式

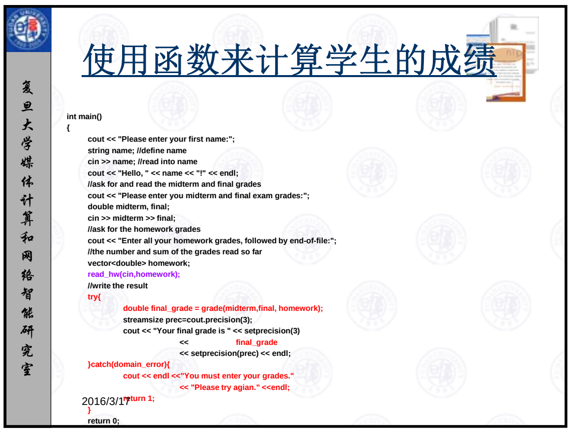
```
double median(vector<double> vec);
```

```
double grade(double midterm, double final, const vector<double> & hw);
```

```
istream & read_hw(istream & in, vector<double> &hw);
```

复制实参，不改变实参的值
不复制实参，不改变实参的值
不复制实参，改变实参的值
传左值

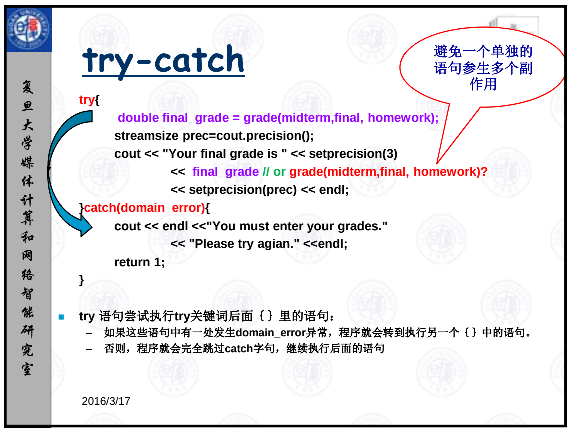
2016/3/17



使用函数来计算学生的成绩

```
int main()
{
    cout << "Please enter your first name:";
    string name; //define name
    cin >> name; //read into name
    cout << "Hello, " << name << "!" << endl;
    //ask for and read the midterm and final grades
    cout << "Please enter you midterm and final exam grades:";
    double midterm, final;
    cin >> midterm >> final;
    //ask for the homework grades
    cout << "Enter all your homework grades, followed by end-of-file:";
    //the number and sum of the grades read so far
    vector<double> homework;
    read_hw(cin,homework);
    //write the result
    try{
            double final_grade = grade(midterm,final, homework);
            streamsize prec=cout.precision(3);
            cout << "Your final grade is " << setprecision(3)
                        <<          final_grade
                        << setprecision(prec) << endl;
    }catch(domain_error){
            cout << endl <<"You must enter your grades."
                        << "Please try agian." <<endl;
            return 1;
    }
    return 0;
```

2016/3/17



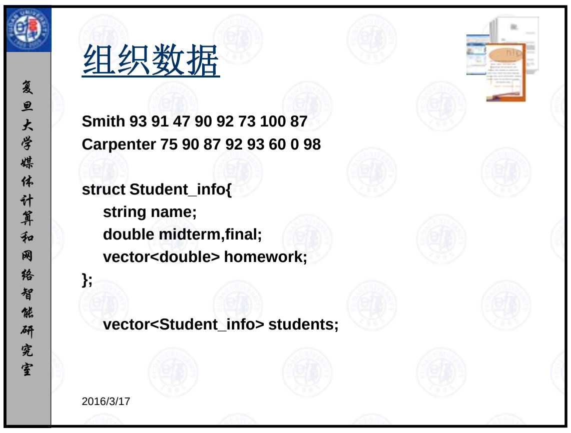
try-catch

```
try{
    double final_grade = grade(midterm,final, homework);
    streamsize prec=cout.precision();
    cout << "Your final grade is " << setprecision(3)
            <<  final_grade // or grade(midterm,final, homework)?
            << setprecision(prec) << endl;
}catch(domain_error){
    cout << endl <<"You must enter your grades."
            << "Please try agian." <<endl;
    return 1;
}
```

避免一个单独的语句参生多个副作用

- try 语句尝试执行try关键词后面 { } 里的语句：
  - 如果这些语句中有一处发生domain_error异常，程序就会转到执行另一个 { } 中的语句。
  - 否则，程序就会完全跳过catch字句，继续执行后面的语句

2016/3/17

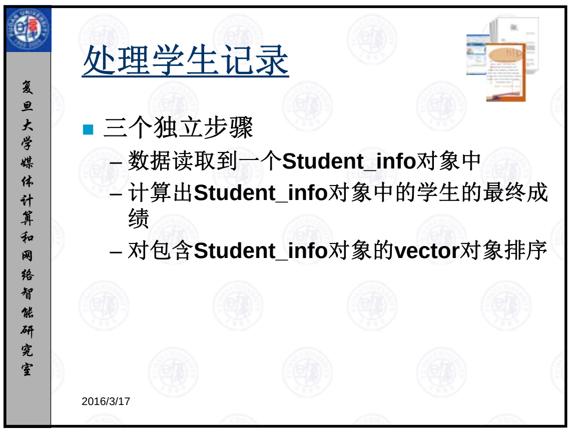


组织数据

Smith 93 91 47 90 92 73 100 87
Carpenter 75 90 87 92 93 60 0 98

```
struct Student_info{
    string name;
    double midterm,final;
    vector<double> homework;
};

    vector<Student_info> students;
```

2016/3/17

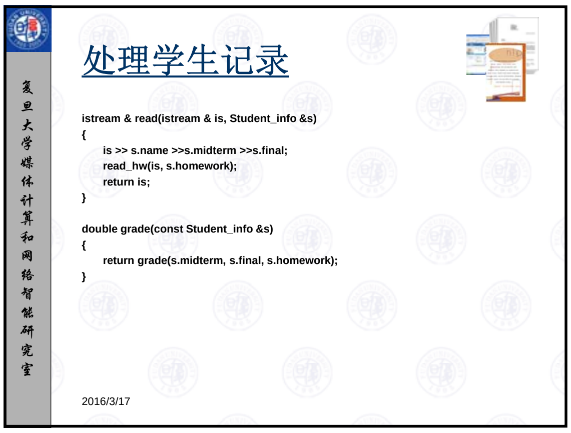


复旦大学媒体计算和网络智能研究室

处理学生记录

- 三个独立步骤
  - 数据读取到一个Student_info对象中
  - 计算出Student_info对象中的学生的最终成绩
  - 对包含Student_info对象的vector对象排序

2016/3/17

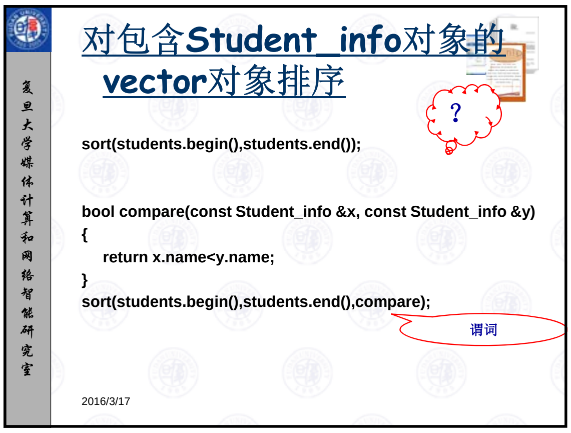


复旦大学媒体计算和网络智能研究室

处理学生记录

```
istream & read(istream & is, Student_info &s)
{
    is >> s.name >>s.midterm >>s.final;
    read_hw(is, s.homework);
    return is;
}

double grade(const Student_info &s)
{
    return grade(s.midterm, s.final, s.homework);
}
```

2016/3/17



复旦大学媒体计算和网络智能研究室

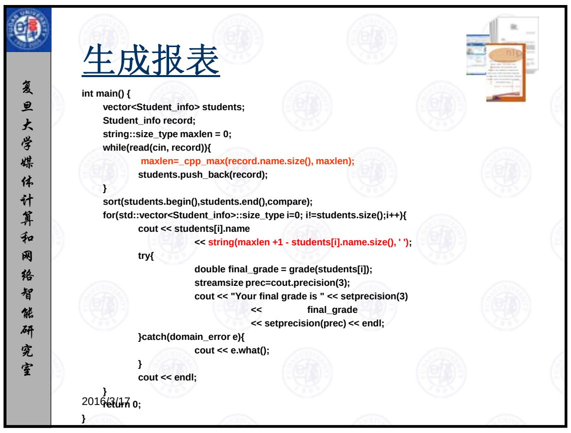
对包含Student_info对象的vector对象排序

?

```
sort(students.begin(),students.end());


bool compare(const Student_info &x, const Student_info &y)
{
    return x.name<y.name;
}
sort(students.begin(),students.end(),compare);
```

谓词

2016/3/17

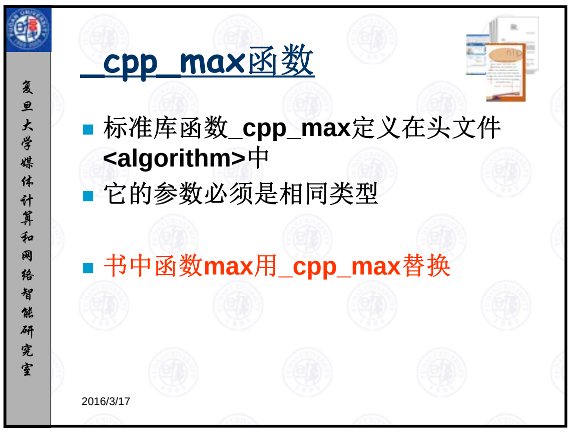


复旦大学媒体计算和网络智能研究室

生成报表

```
int main() {
    vector<Student_info> students;
    Student_info record;
    string::size_type maxlen = 0;
    while(read(cin, record)){
            maxlen=_cpp_max(record.name.size(), maxlen);
            students.push_back(record);
    }
    sort(students.begin(),students.end(),compare);
    for(std::vector<Student_info>::size_type i=0; i!=students.size();i++){
            cout << students[i].name
                    << string(maxlen +1 - students[i].name.size(), ' ');
            try{
                    double final_grade = grade(students[i]);
                    streamsize prec=cout.precision(3);
                    cout << "Your final grade is " << setprecision(3)
                            <<          final_grade
                            << setprecision(prec) << endl;
            }catch(domain_error e){
                    cout << e.what();
            }
            cout << endl;
    }
    return 0;
}
```

2016/3/17



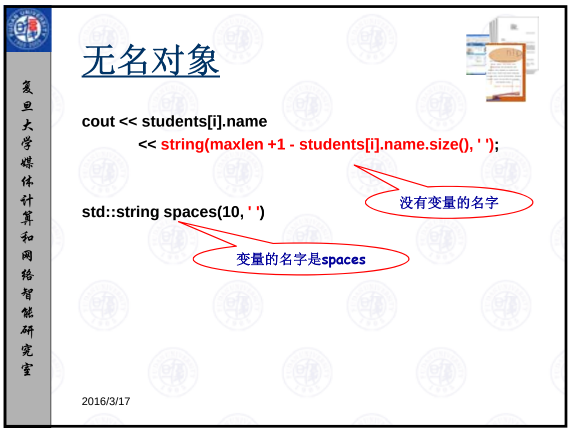
复旦大学媒体计算和网络智能研究室

_cpp_max函数

- 标准库函数_cpp_max定义在头文件<algorithm>中
- 它的参数必须是相同类型

- 书中函数max用_cpp_max替换

2016/3/17



复旦大学媒体计算和网络智能研究室

无名对象

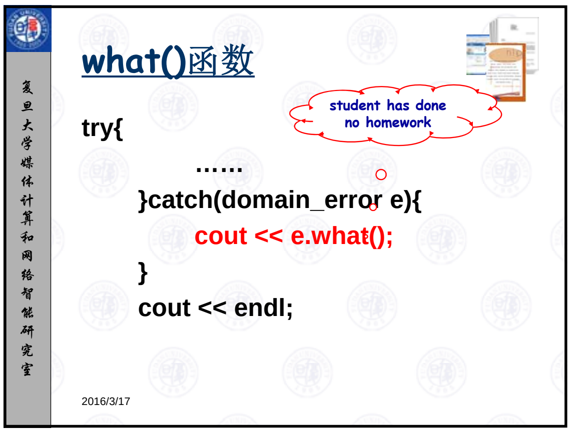
```
cout << students[i].name
        << string(maxlen +1 - students[i].name.size(), ' ');
```

没有变量的名字

```
std::string spaces(10, ' ')
```

变量的名字是spaces

2016/3/17



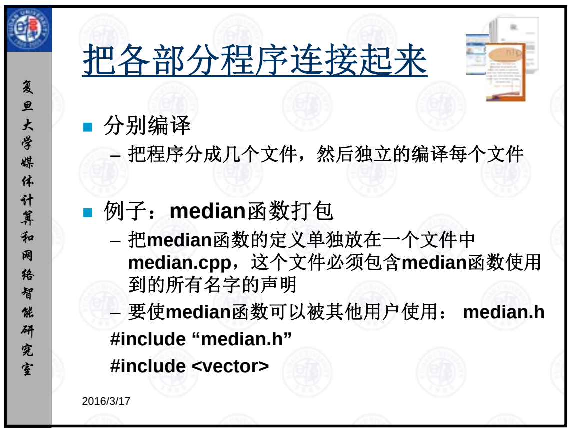
what()函数

复旦大学媒体计算和网络智能研究室

student has done no homework

```
try{
        ……
    }catch(domain_error e){
        cout << e.what();
    }
    cout << endl;
```

2016/3/17



把各部分程序连接起来

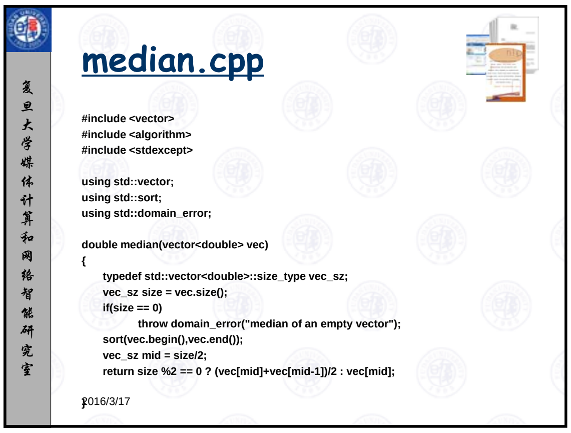
复旦大学媒体计算和网络智能研究室

- 分别编译
  - 把程序分成几个文件，然后独立的编译每个文件
- 例子：median函数打包
  - 把median函数的定义单独放在一个文件中median.cpp，这个文件必须包含median函数使用到的所有名字的声明
  - 要使median函数可以被其他用户使用： median.h

#include “median.h”

#include <vector>

2016/3/17

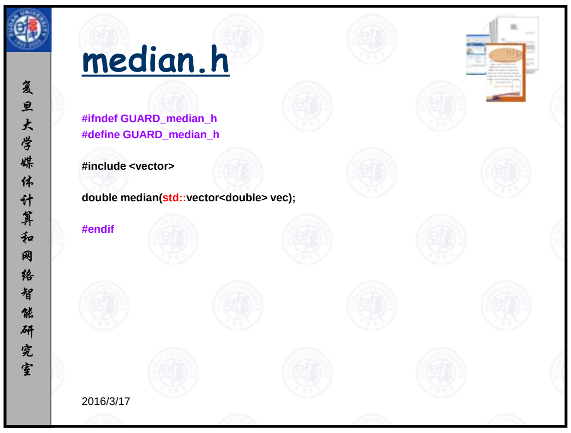


median.cpp

复旦大学媒体计算和网络智能研究室

```
#include <vector>
#include <algorithm>
#include <stdexcept>

using std::vector;
using std::sort;
using std::domain_error;

double median(vector<double> vec)
{
    typedef std::vector<double>::size_type vec_sz;
    vec_sz size = vec.size();
    if(size == 0)
            throw domain_error("median of an empty vector");
    sort(vec.begin(),vec.end());
    vec_sz mid = size/2;
    return size %2 == 0 ? (vec[mid]+vec[mid-1])/2 : vec[mid];
}
```

2016/3/17

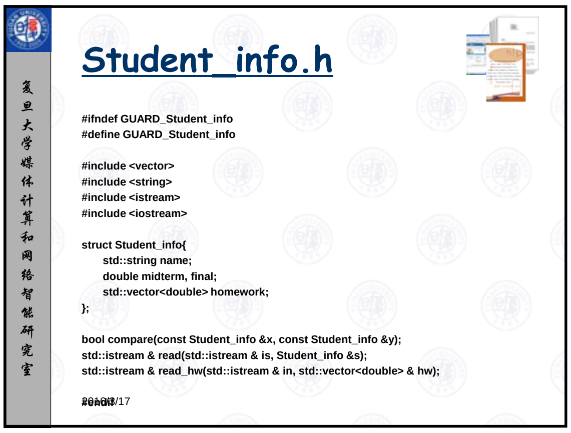


median.h

复旦大学媒体计算和网络智能研究室

```
#ifndef GUARD_median_h
#define GUARD_median_h

#include <vector>

double median(std::vector<double> vec);

#endif
```

2016/3/17

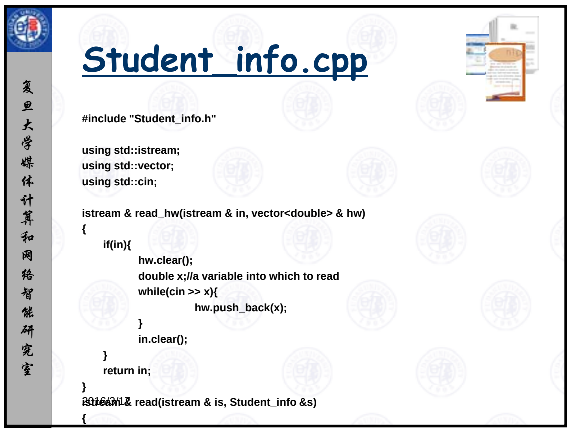


Student_info.h

复旦大学媒体计算和网络智能研究室

```
#ifndef GUARD_Student_info
#define GUARD_Student_info

#include <vector>
#include <string>
#include <istream>
#include <iostream>

struct Student_info{
    std::string name;
    double midterm, final;
    std::vector<double> homework;
};

bool compare(const Student_info &x, const Student_info &y);
std::istream & read(std::istream & is, Student_info &s);
std::istream & read_hw(std::istream & in, std::vector<double> & hw);

#endif
```

2016/3/17



Student_info.cpp

复旦大学媒体计算和网络智能研究室

```
#include "Student_info.h"

using std::istream;
using std::vector;
using std::cin;

istream & read_hw(istream & in, vector<double> & hw)
{
    if(in){
            hw.clear();
            double x;//a variable into which to read
            while(cin >> x){
                    hw.push_back(x);
            }
            in.clear();
    }
    return in;
}
istream & read(istream & is, Student_info &s)
{
```

2016/3/17

复旦大学媒体计算和网络智能研究室

grade.h

```
#ifndef GUARD_grade_h
#define GUARD_grade_h

#include <vector>
#include "Student_info.h"

double grade(double midterm, double final, double homework);
double grade(double midterm, double final, const std::vector<double> & hw);
double grade(const Student_info &s);

#endif
```

2016/3/17

复旦大学媒体计算和网络智能研究室

grade.cpp

```
#include <vector>
#include <stdexcept>
#include "grade.h"
#include "median.h"

using std::vector;
using std::domain_error;

double grade(double midterm, double final, double homework)
{
    return 0.2 * midterm + 0.4 * final + 0.4 * homework;
}

double grade(double midterm, double final, const vector<double> & hw)
{
    if(hw.size()==0)
        throw domain_error("student has done no homework");
    return grade(midterm, final, median(hw));
}
```

2016/3/17

复旦大学媒体计算和网络智能研究室

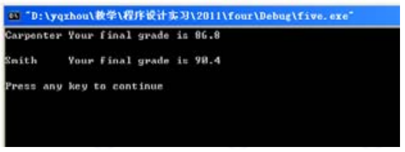
main.cpp

```
#include <iostream>
#include <ios>
#include <string>
#include <iomanip>
#include <vector>
#include <stdexcept>
#include <algorithm>
#include "grade.h"
#include "Student_info.h"

using std::cin;                         using std::setprecision;
using std::cout;    using std::string;
using std::endl;    using std::streamsize;
using std::vector;  using std::sort;
using std::domain_error; using std::_cpp_max;

int main()
{
        vector<Student_info> students;
        Student_info record;
        string::size_type maxlen = 0;

        while(read(cin, record)){
                        maxlen=_cpp_max(record.name.size(), maxlen);
                        students.push_back(record);
        }
        sort(students.begin(),students.end(),compare);

        for(std::vector<Student_info>::size_type i=0; i!=students.size();i++){
                        cout << students[i].name
                                << string(maxlen +1 - students[i].name.size(), ' ');

                        try{
                                double final_grade = grade(students[i]);
                                streamsize prec=cout.precision(3);
                                cout << "Your final grade is " << setprecision(3)
                                                << final_grade
                                                << setprecision(prec) << endl;
```

2016/3/17

复旦大学媒体计算和网络智能研究室

程序输出



```
"D:\yqzhou\教学\程序设计实习\2011\four\Debug\five.exe"
Carpenter Your final grade is 86.8

Smith     Your final grade is 90.4

Press any key to continue
```

2016/3/17