

简单实例

- 设计一个类Student，用于记录学生的姓名、学号、成绩等数据，具有计算学生个人平均成绩、计算个人学生绩点、输出学生基本信息等功能，并且可以和给定的测试主程序配合使用。
- 书面作业：Class Student 1.0版本
- 提交时间：2019年5月29日



测试主程序

```
#include "iostream"
using namespace std;
#include "string"
int main()
{ Cstudent Stu1(303001, "zhang", 'M', 21);
  Cstudent Stu2(Stu1);
  Cstudent Stu3;
  string name;

  name=Stu1.getname();
  Stu1.inputscores();
  Stu1.average_score();
  Stu1.grade_compute();
  Stu1.displayinfo();
```

```
Stu2.ResetInfo(303002, "LI", 'F',20);  
Stu2.inputscores();  
Stu2.average_score();  
Stu2.grade_compute();  
Stu2.displayinfo();
```

```
Stu3.SetID(303003);  
Stu3.SetName("LI");  
Stu3.SetGender('F');  
Stu3.SetAge(22);  
Stu3.inputscores();  
Stu3.average_score();  
Stu3.grade_compute();  
Stu3.displayinfo();
```

```
return 0;
```

```
}
```



Student类的设计

- 数据成员

- ID long
- name string
- age int
- Gender char
- float scores[10];
- float averagescore;
- float gradescore;
- 全部是私有属性

- 成员函数

- 构造函数（无参、有四个参数、复制构造）
- 析构函数
- 功能函数
 - 成绩输入inputscores
 - 平均分计算averagescores
 - 绩点计算grade_compute
 - 基本信息输出displayinfo
 - 个人信息重置ResetInfo
 - 信息设置：SetName等
 - 信息获取：GetName等
- 全部是公有

Student类的定义

```
class Cstudent
```

```
{ private:
```

```
    long ID;
```

```
    string name;
```

```
    int age;
```

```
    char gender;
```

```
    float scores[6];
```

```
    float averagescore;
```

```
    float gradescore;
```

```
public:
```

```
    Cstudent();
```

```
    Cstudent(long stuid,
```

```
                string stuname,
```

```
                char stugender,
```

```
                int stuage);
```

```
    Cstudent(const Cstudent &stu);
```

```
    ~Cstudent(){;}
```

```
    void SetName(string stuname);
```

```
    string getname();
```

```
    ...
```

```
    float average_scores();
```

```
    void displayinfo();
```

```
}
```

Student类成员函数的实现

- 复制构造函数

```
Cstudent::Cstudent  
    (const Cstudent &stu);  
{ ID = stu.ID;  
  name = stu.name;  
  age = stu.age;  
  gender = stu.gender;  
}
```

- 基本信息输出函数

```
void Cstudent::displayinfo()  
{ cout<<ID<<" "  
  <<name<<" "  
  <<gender<<" "  
  <<age<<" "  
  <<averagescore<<" "  
  <<gradescore<<endl;  
  cout<<"detail scores:"  
  for ( int i=0; i<6; i++)  
      cout<<scores[i]<<" ";  
  cout<<endl;  
}
```

Student类的扩展实现

- 对象数组如何应用
 - 一个年级有300个学生，需要创建一个CS_student的对象数组
 - 如何创建？
 - Student类的构造函数是否需要调整？
 - 对象数组中的对象如何创建？
 - 自行给出main测试程序
- 书面作业：Class Student 2.0版本
- 提交时间：2019年5月31日



Student类的扩展实现

- 计算每生的绩点时，需要知道每门课程的学分
- 需要一个学分数组
- 学分数组并不属于每一个学生，如何定义
 - 静态数据成员
- 请完成学分数组的定义
- 请完成学分数组的输入和查询
 - 静态成员函数
- 完成Student类中绩点计算函数
- 书面作业：Class Student 3.0版本
- 提交时间：2019年5月31日

Student类的扩展实现

- main函数中需要计算年级中每门课程的平均分，最高分，最低分，调用以下三个函数来实现上述功能
 - average
 - max
 - min
- 三个函数如何定义？
 - 三个函数需要访问CS_student[300]这个对象数组中每个对象中的私有数据成员scores[11]数组
 - 三个函数一定是友元函数
 - 请参照前面average函数完成Student类的三个友元函数：max、min和average
- 书面作业：Class Student 4.0版本
- 提交时间：2019年6月5日

Student类的扩展

- Student对象需要根据绩点进行排序
 - 通常需要进行学生绩点成绩的大小比较
 - 能否根据学生绩点成绩重载比较运算符并实现一个排序算法
- 书面作业：Class Student 5.0版本
- 提交时间：2019年6月12日

