面向对象：类和对象

C语言：面向过程

C++：基于面向对象的

面向过程 & 面向对象

面向过程：做事情的具体的步骤(开车例子)

1.开车门

2.上车

3.关车门

4.系安全带

5.打火

6.挂挡

7.放手刹

8.出发

面向对象：

人　　车

名字、年龄　　技能　　参数信息　　功能

描述人的群体：

面向过程：--->蛋炒饭

面向对象：--->盖浇饭

在C语言的结构体中不可以定义函数，而在C++的结构体(类)中是可以的

在C语言中用结构体定义出来的叫做变量，在C++中使用结构体定义出来的叫做一个实体/对象

```
// 类
struct Student{
```

```
    void SetStudentInfo(char *name, char *gender, int age){
        strcpy(_name, name);
        strcpy(_gender, gender);
        _age = age;
    }
    void PrintStudentInfo(){
        cout << _name << " " << _gender << " " << _age << endl;
    }
    char _name[20];
    char _gender[3];
    int _age;
};
int main() {
    Student s1, s2, s3;
    s1.SetStudentInfo("Peter", "男", 18);
    s2.SetStudentInfo("jing jing", "女", 17);
    s3.SetStudentInfo("summer", "男", 2);

    s1.PrintStudentInfo();
    s2.PrintStudentInfo();
    s3.PrintStudentInfo();
    return 0;
}
```

封装的特性：

封装的概念：

函数：将一些语句按照一定的逻辑顺序包装在一起

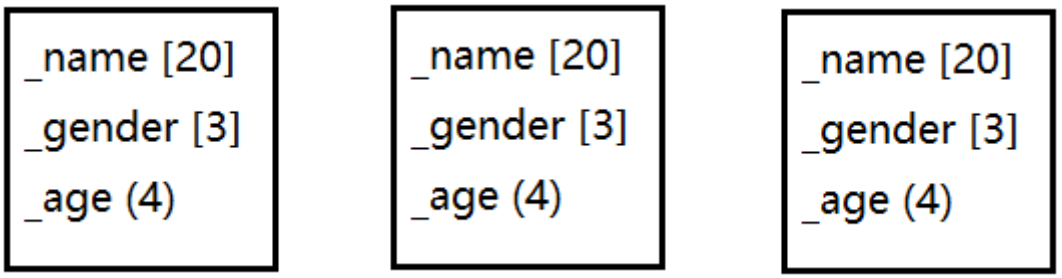
笔记本：

C++封装：类 -- 可以将对象的属性和方法包装在一起

访问限定符：private、public、protected

类 -- 类型 ----> 实例化 ----> 对象

图纸 ----------> 建造 ------> 别墅



类的打大小：将类中的成员变量加起来，结合内存对齐

编译器识别类的步骤：

1. 识别类名
2. 识别类中成员
3. 识别类中的成员函数 & 队成员函数进行改写

```
struct Student{
    /*
    void SetStudentInfo(Student* const this, char *name, char *gender, int age) {
        strcpy(this->_name, name);
        strcpy(this->_gender, gender);
        this->_age = age;
    }
    */
    void SetStudentInfo(char *name, char *gender, int age){
        strcpy(_name, name);
```

```
        strcpy(_gender, gender);
        _age = age;
    }
    /*
    void PrintStudentInfo(Student *this){
        count << this->_name << " " << this->_gender << " " <<
this->_age << endl;
    }
    */
    void PrintStudentInfo(){
        cout << _name << " " << _gender << " " << _age << endl;
    }
    char _name[20];
    char _gender[3];
    int _age;
};
int main() {
    Student s1;
    s1.SetStudentInfo("Peter", "男", 18);
    // Student::setStudentInfo(&s, "Perter", "男", 18);

    s1.PrintStudentInfo();
    // Student::PrintStudentInfo(&s);
    return 0;
}
```