



华中科技大学

# 面向对象程序设计

许向阳

**xuxy@hust.edu.cn**



# 教材

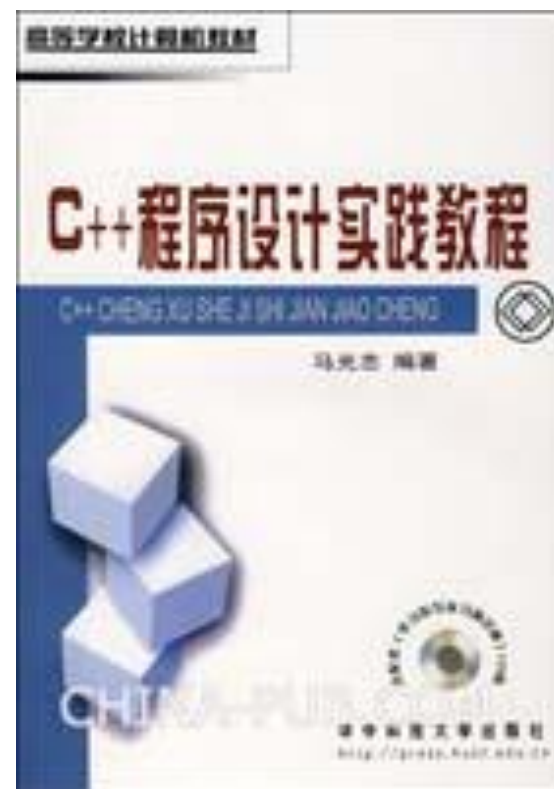


华中科技大学

教材：C++程序设计实践教程

出版：华中科技大学出版社

编著：马光志





# 参考书

- [1] 张德慧. C++面向对象程序设计.  
科学出版社, 2005年
- [2] 沈学东. C++面向对象程序设计实用教程.  
上海交通大学出版社, 2012年
- [3] C++编程思想 Thinking in C++  
刘宗田等译, 机械工业出版社
- [4] C++ Primer, 李师贤等译, 人民邮电出版社
- [5] C++高效编程, Effect C++
- [6] C++编程语言, C++ Program Language
- [7] C++入门经典 The C++ Language Beginning



# 学习群



华中科技大学



群名称：华科大计算机C++...

群 号：477368987





# 第一章 宏论面向对象的程序设计

- 什么是面向对象的程序设计？
- 为什么要面向对象的程序设计？
- 如何学习这门课？



# 什么是面向对象？

## 人类的思维方式：面向对象

- 客观世界有很多各种各样的对象组成
  - 每种对象有各自的内部状态和运动规律
  - 不同的对象之间相互作用和联系
  - 对象是组成系统的基本逻辑单元
  - 对象可表示抽象的概念、具体的模块
- 面向对象的思想贯穿软件开发的整个过程



# 什么是面向对象？



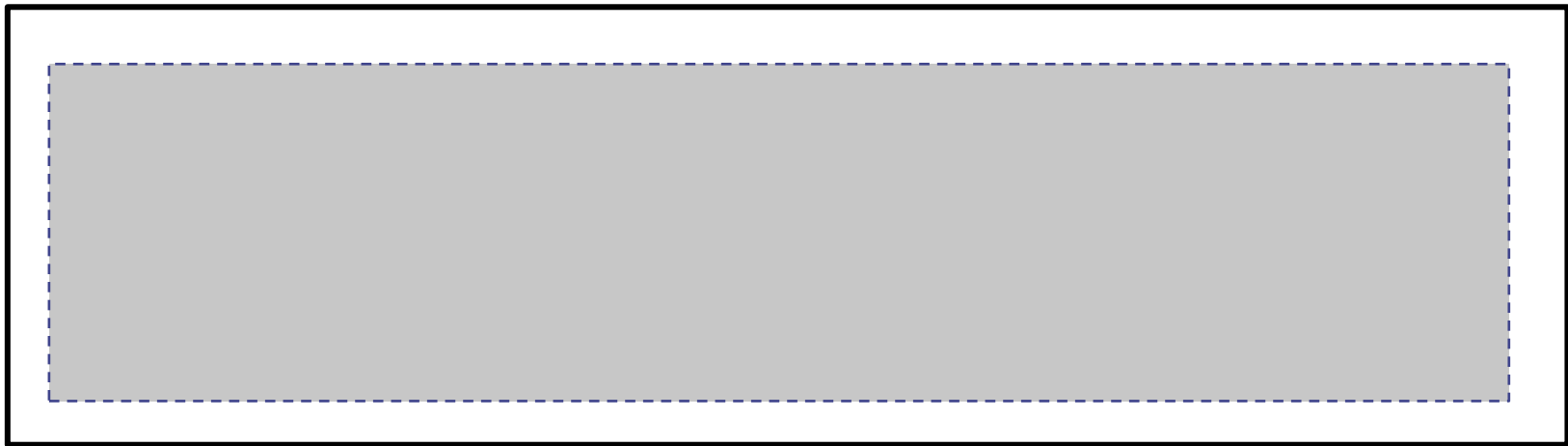


# 什么是面向对象？





# 什么是面向对象？





# 什么是面向对象？

## 种菜

### 种菜

菜地  
菜的状态  
浇水  
施肥  
治虫

## 配菜

### 配菜

菜、净菜  
  
择菜  
洗菜  
切菜

## 炒菜

### 炒菜

净菜  
油、盐...  
  
点火  
下锅

分工、交接（接口）  
信息隐藏





# 什么是面向对象？

分工协作

加工对象与加工工作捆绑

信息隐藏





# 什么是面向对象？

拍一部电影，反映菜从田间地头到餐桌的过程

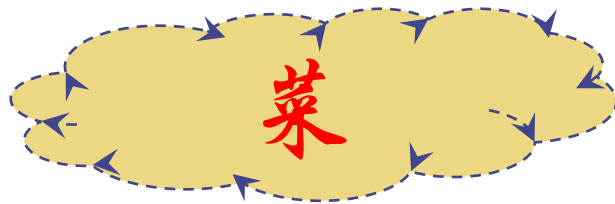
信息的组织及传递，谁占据画面的重要位置？

角色：菜农、商贩、择菜工、厨师、食客

场地：菜地、菜场、厨房、餐厅



# 什么是面向对象？



- 不同人说“菜”时，各指的是什么意思？  
菜农、商贩、择菜工、厨师、食客
- 不同场景下说“菜”时，各指的是什么意思？  
菜地、菜场、厨房、餐厅

多 态



# 软件开发的一般过程是什么？

任务：开发一个高校的信息管理系统

## □ 需求分析

用户需要的是什么？

怎样清晰的展现用户的需求？

## □ 系统设计

总体结构、功能模块、数据结构、接口

## □ 系统实现

编写程序、调试程序、优化程序

## □ 系统维护

系统升级





# 如何更好的设计开发软件？

任务：开发一个高校的信息管理系统

- 需求分析

面向对象的分析，统一建模语言UML

- 系统设计

面向对象的设计

- 系统实现

面向对象的实现：C++，Java

- 系统维护

更好是全方位的、贯穿分析、设计、实现







# 什么是面向对象？

任务：开发一个高校的信息管理系统

学生管理、教师管理、科研管理、设备管理.....

学生（Student）

姓名、学号、身份证号

.....

输入学生信息  
修改学生信息  
查询学生信息

.....

课程（Course）

课名、学时、教材名称

.....

输入课程信息  
修改课程信息  
查询课程信息

.....



# 什么是面向对象？

任务：开发一个高校的信息管理系统



对象 与对象之间相互关联



# 面向对象的核心是什么？

面向对象的核心

封装 (Encapsulation)

继承 (Inheritance)

多态 (Polymorphism)



# 面向对象的核心是什么？

## 封装

- C语言中定义结构 struct 就是封装
- struct结构中封装了什么？
- 封装有何好处？
- 面向对象的封装又增加了些什么？

程序长度超过5万行，名字冲突就成了一个重要问题。





# 面向对象的核心是什么？

## 封装

- 将对象的**特征**和**行为**包装在一起
- **特征**描述      对象的“组织结构”      **数据成员**
- **行为**描述对象的“功能”      **函数成员**
- 封装定义了对象的边界
- 提供了外部访问的接口
- 屏蔽了对象内部的特征、行为细节

不同的对象中，可以有同名的函数





# 面向对象的核心是什么？

## 继承

- 一个对象获得另一个或多个对象的**特征和行为**
- 小孩长相象父母是获得父母“特征”
- 走路象父亲是获得父亲“行为”



# 面向对象的核心是什么？

## 继承

任务：开发一个高校  
信息管理系统

□ 学生管理：对象学生

□ 教师管理：对象教师

定义新的对象：人 (Person)

特征：姓名、身份证号、联系电话 .....

行为：录入电话号码、修改电话号码.....

学生、教师：都继承 Person 的特征和行为

同时，又各自新增一些自己的特征和行为







# 面向对象的核心是什么？

## 多态

**多态：**调用同名函数，表现出的不同行为

**静态多态：**编译时明确了同名函数中的哪一个

**动态多态：**运行时才确定调用同名函数中的哪一个

在一个对象中，可以有同名的函数

**VS**

在不同对象中，可以有同名的函数





# 面向对象的核心是什么？

## 静态多态

实现两个数相加返回结果的加法函数

```
int  fadd(int x, int y);  
double  fadd(double x, double y);  
void  fadd(int *z, int x, int y);
```

```
int  x,y,z;      double u,v,w;  
z=fadd(x,y);  
w=fadd(u,v);  
fadd(&z, x,y);
```





# 面向对象的核心是什么？

## 静态多态

如何区分同名函数？

C++编译器的换名机制

区分参数个数不同或者参数类型不同的函数，  
若只是返回值类型不同，则不能区分

**PUBLIC ?fadd@@YAHHH@Z**

**PUBLIC ?fadd@@YAXPAHHH@Z**

**PUBLIC ?fadd@@YANNN@Z**





# 面向对象的核心是什么？

## 动态多态

**Fruit : BuyFruit**

苹果摊前：老板，水果怎么卖？

老板： **BuyFruit** {这是红富士苹果， **10元**  
一斤}

梨子摊前：老板，水果怎么卖？

老板： **BuyFruit** {...梨子**5元**一斤}

香蕉摊前：老板，水果怎么卖？

老板： **BuyFruit** {...便宜呀， **10元3斤**}





# 面向对象的核心是什么？

## 动态多态

```
APPLE a;   PEAR  p;   BANANA b;  
FRUIT *q;  
q=&a;  or  q=&p;  or  q=&b;  
q->BuyFruit();
```

如何实现动态多态？

从机器语言的角度来看，子程序调用有哪些方式？

CALL ???





# 面向对象的核心是什么？

## 解决C程序语言中存在的问题

特征与行为分离，封装度不够

➤ 特征的封装，不同struct中可以有同名变量

➤ 对象的封装，不同对象中可以有同名函数

特征与行为整合在一起

不同函数实现相似功能，重用、共享度不高

➤ 对象的继承，重用父类对象的特征和行为

相同功能的函数，用不同的名称，自适应程度不高

➤ 多态：静态多态、动态多态





# 为什么要学面向对象技术？

- 面向对象的技术是为了**更好地**开发软件
- **全方位**支持系统分析、设计、实现
- 核心特征是封装、继承、多态





# 其他

➤ C++与面向对象的程序设计关系是什么？  
C++、Java、C#

➤ C++与C语言的关系是什么？

C++兼容C，支持原有的C语法

C++ 混合支持结构化和面向对象





华中科技大学

# 如何学习这门课？

- 建构主义理论
- 问题牵引
- 理解原理
- 实践、用心





# 如何学习这门课？

## 用心体会差异

屡战屡败 VS 屡败屡战

一切为了学生、为了一切学生、为了学生一切

辣不怕、不怕辣、怕不辣

减肥 VS 瘦身

指针数组 VS 数组指针

指针常量 VS 常量指针





# 如何学习这门课?

## ➤ 用心 程序设计中

命名及模块划分

可读性

防御式编程

鲁棒性

用户输入数据时, 不符合事实(错误输入)

函数调用实参不满足函数实现体假设的参数要求

运行速度

高效性





# 如何学习这门课?

## 运行速度

## 高效性

```
int a[10000][40000]; 将所有元素都置为0
for (i = 0; i < 10000; i++)           // 方法1
    for (j = 0; j < 40000; j++)
        a[i][j] = 0;
```

```
for (j = 0; j < 40000; j++)           // 方法2
    for (i = 0; i < 10000; i++)
        a[i][j] = 0;
```

```
memset(a, 0, sizeof(int)*10000 * 40000); // 方法3
```





华中科技大学

