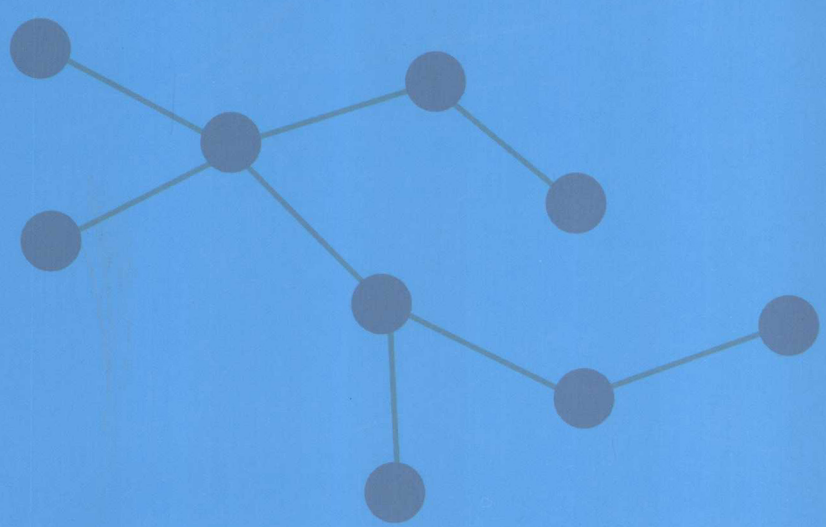
高等院校计算机任务驱动教改教材

C++程序设计

王艳娟　郭　龙　徐书海　主　编
王顺强　崔　敏　王　娟　副主编



清华大学出版社

高等院校计算机任务驱动教改教材

C++程序设计

王艳娟　郭　龙　徐书海　主　编
王顺强　崔　敏　王　娟　副主编

清华大学出版社
北京

内 容 简 介

本书是C++程序设计基础教材。全书共有6大模块，主要内容涉及C++基础知识、类与对象、继承与派生、运算符重载、多态性与虚函数，以及I/O流与文件。本书内容简单易懂，重点突出，深入浅出，用大量的实例来解析和阐明C++语言的基本原理。

本书适合作为本科、专科学生学习C++程序设计的教材，也可以作为计算机爱好者的参考用书。

图书在版编目(CIP)数据

C++程序设计/王艳娟，郭龙，徐书海主编. --北京：清华大学出版社，2016
高等院校计算机任务驱动教改教材
ISBN 978-7-302-44521-0

Ⅰ. ①C… Ⅱ. ①王… ②郭… ③徐… Ⅲ. ①C语言－程序设计－高等学校－教材 Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2016)第171797号

责任编辑：张龙卿
封面设计：徐日强
责任校对：刘　静
责任印制：沈　露

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦A座　　**邮　　编**：100084
社 总 机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课件下载：http://www.tup.com.cn，010-62770175-4278
印 装 者：北京嘉实印刷有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：10.5　　**字　　数**：236千字
版　　次：2016年8月第1版　　**印　　次**：2016年8月第1次印刷
印　　数：1～2000
定　　价：26.00元

产品编号：060822-01

编审委员会

出版说明

我国高职高专教育经过十几年的发展，已经转向深度教学改革阶段。教育部于2006年12月发布了教高[2006]第16号文件《关于全面提高高等职业教育教学质量的若干意见》，大力推行工学结合，突出实践能力培养，全面提高高职高专教学质量。

清华大学出版社作为国内大学出版社的领跑者，为了进一步推动高职高专计算机专业教材的建设工作，适应高职高专院校计算机类人才培养的发展趋势，根据教高[2006]第16号文件的精神，2007年秋季开始了切合新一轮教学改革的教材建设工作。该系列教材一经推出，就得到了很多高职院校的认可和选用，其中部分书籍的销售量都超过了3万册。现重新组织优秀作者对部分图书进行改版，并增加了一些新的图书品种。

目前国内高职高专院校计算机网络与软件专业的教材品种繁多，但符合国家计算机网络与软件技术专业领域技能型紧缺人才培养培训方案，并符合企业的实际需要，能够自成体系的教材还不多。

我们组织国内对计算机网络和软件人才培养模式有研究并且有过一段实践经验的高职高专院校，进行了较长时间的研讨和调研，遴选出一批富有工程实践经验和教学经验的双师型教师，合力编写了这套适用于高职高专计算机网络、软件专业的教材。

本套教材的编写方法是以任务驱动、案例教学为核心，以项目开发为主线。我们研究分析了国内外先进职业教育的培训模式、教学方法和教材特色，消化吸收优秀的经验和成果。以培养技术应用型人才为目标，以企业对人才的需要为依据，把软件工程和项目管理的思想完全融入教材体系，将基本技能培养和主流技术相结合，课程设置中重点突出、主辅分明、结构合理、衔接紧凑。教材侧重培养学生的实战操作能力，学、思、练相结合，旨在通过项目实践，增强学生的职业能力，使知识从书本中释放并转化为专业技能。

一、教材编写思想

本套教材以案例为中心，以技能培养为目标，围绕开发项目所用到的知识点进行讲解，对某些知识点附上相关的例题，以帮助读者理解，进而将知识转变为技能。

考虑到是以“项目设计”为核心组织教学，所以在每一学期配有相应的实训课程及项目开发手册，要求学生在教师的指导下，能整合本学期所学的知识内容，相互协作，综合应用该学期的知识进行项目开发。同时，在教材中采用了大量的案例，这些案例紧密地结合教材中的各个知识点，循序渐进，由浅入深，在整体上体现了内容主导、实例解析、以点带面的模式，配合课程后期以项目设计贯穿教学内容的教学模式。

软件开发技术具有种类繁多、更新速度快的特点。本套教材在介绍软件开发主流技术的同时，帮助学生建立软件相关技术的横向及纵向的关系，培养学生综合应用所学知识的能力。

二、丛书特色

本系列教材体现目前工学结合的教改思想，充分结合教改现状，突出项目面向教学和任务驱动模式教学改革成果，打造立体化精品教材。

(1) 参照和吸纳国内外优秀计算机网络、软件专业教材的编写思想，采用本土化的实际项目或者任务，以保证其有更强的实用性，并与理论内容有很强的关联性。

(2) 准确把握高职高专软件专业人才的培养目标和特点。

(3) 充分调查研究国内软件企业，确定了基于 Java 和. NET 的两个主流技术路线，再将其组合成相应的课程链。

(4) 教材通过一个个的教学任务或者教学项目，在做中学，在学中做，以及边学边做，重点突出技能培养。在突出技能培养的同时，还介绍解决思路和方法，培养学生未来在就业岗位上的终身学习能力。

(5) 借鉴或采用项目驱动的教学方法和考核制度，突出计算机网络、软件人才培训的先进性、工具性、实践性和应用性。

(6) 以案例为中心，以能力培养为目标，并以实际工作的例子引入概念，符合学生的认知规律。语言简洁明了、清晰易懂，更具人性化。

(7) 符合国家计算机网络、软件人才的培养目标；采用引入知识点、讲述知识点、强化知识点、应用知识点、综合知识点的模式，由浅入深地展开对技术内容的讲述。

(8) 为了便于教师授课和学生学习，清华大学出版社正在建设本套教材的教学服务资源。在清华大学出版社网站(www. tup. com. cn)免费提供教材的电子课件、案例库等资源。

高职高专教育正处于新一轮教学深度改革时期，从专业设置、课程体系建设到教材建设，依然是新课题。希望各高职高专院校在教学实践中积极提出意见和建议，并及时反馈给我们。清华大学出版社将对已出版的教材不断地修订、完善，提高教材质量，完善教材服务体系，为我国的高职高专教育继续出版优秀的高质量的教材。

清华大学出版社

高职高专计算机任务驱动模式教材编审委员会

2014 年 3 月

前　言

面向对象的程序设计语言 C++ 是一门非常重要的计算机专业基础课程，学习 C++ 不仅可以深入理解面向对象程序设计的基本思想，而且有助于很多后续课程的学习。

封装、继承和多态是面向对象语言的三大基本特性，为此在 C++ 程序设计课程中，将面向对象的概念、构造函数、虚基类以及多态作为重点和难点，同时这也是很多学生学习时的主要障碍。本书在编写的过程中由浅入深，从最基本的理论出发，采用大量的实例来阐明相关知识，减轻了初学者的学习困难。

本书共有 6 大模块，模块 1 为 C++ 基础知识，模块 2 为类与对象，模块 3 为继承与派生，模块 4 为运算符重载，模块 5 为多态性与虚函数，模块 6 为 I/O 流与文件。

本书由王艳娟、郭龙、徐书海担任主编，王顺强、崔敏、王娟担任副主编。王艳娟编写模块 1 至模块 4，郭龙、徐书海编写模块 5，王顺强、崔敏、王娟编写模块 6。全书由王艳娟统稿。

由于编者水平有限，书中难免有错误和不妥之处，敬请读者批评指正。

编者

2016 年 4 月

目 录

模块 1　C++ 基础知识

案例引入

编写程序实现以下功能：求取两个整型数据、两个字符型数据以及两个实型数据的最大值。

在 C 语言中有明确的规定，多个函数不能使用同一个函数名，为此，在 C 语言中的具体实现如下。

```
#include<stdio.h>
int maxint(int a,int b)                    //求取两个整数的最大值函数
{
  int max;
  if(a>b)
    max=a;
  else
    max=b;
  return max;
}
char maxchar(char a,char b)                //求取两个字符的最大值函数
{
  char max;
  if(a>b)
    max=a;
  else
    max=b;
  return max;
}
double maxfloat(double a, double b)        //求取两个实数的最大值函数
{
  double max;
  if(a>b)
    max=a;
  else
    max=b;
  return max;
}
void main()
{
  printf("%d\t%c\t%f\n",maxint(5,8),maxchar('h','p'),maxfloat(7.9,90.8));
}
```

从上述代码中可以看出：在 C 语言中，需要定义三个函数，分别实现求取两个数最大值的功能，且三个函数的名字不能相同，同时我们还发现三个函数中除了使用的数据类型不同，其余具体的实现代码均是一样的。

C++ 中如何实现呢？会不会也要这么麻烦呢？答案是在 C++ 中实现求取两个数的最大值可以有如下两种不同的方式。

(1) 使用相同函数名定义三个函数。

```
#include<iostream>                          //包含输入输出流头文件
using namespace std;                        //引入标准化命名空间
int max(int a,int b)
{
  int t;
  if(a>b)
    t=a;
  else
    t=b;
  return t;
}
char max(char a,char b)
{
  char t;
  if(a>b)
    t=a;
  else
    t=b;
  return t;
}
double max(double a,double b)
{
  float t;
  if(a>b)
    t=a;
  else
    t=b;
  return t;
}
void main()
{
  cout<<max(5,8)<<endl;
  cout<<max('h','p')<<endl;
  cout<<max(7.9,90.8)<<endl;
}
```

(2) 使用一个函数实现。

```
#include<iostream>                          //包含输入输出流头文件
using namespace std;                        //引入标准化命名空间
```

```
template<typename T>                    //模板的定义
T max(T a,T b)
{
  T t;
  if(a>b)
    t=a;
  else
    t=b;
  return t;
}
void main()
{
  cout<<max(5,8)<<endl;
  cout<<max('h','p')<<endl;
  cout<<max(7.9,90.8)<<endl;
}
```

C++ 中上述代码是如何实现的呢？在程序运行期间有三个相同的函数名，如何进行识别并正确调用相关的函数呢？只有一个函数的定义，如何实现三种不同类型数据最大值的求取？T 又代表什么？又如何确定是整型、字符型还是实型的呢？下面我们进入 C++ 的世界去慢慢探索吧！

1.1　面向对象程序设计的基本思想

1.1.1　C++ 的由来

C 语言是结构化和模块化的语言，是面向过程的。在处理小规模的程序时，运用 C 语言较为得心应手，但是当问题比较复杂、程序规模比较大时，结构化的程序设计方法就显示出它的不足。C 语言的设计者必须细致地设计程序中的每一个细节，准确地考虑程序运行时所发生的事情。为了解决软件设计危机，在 20 世纪 80 年代提出了面向对象的程序设计思想(Object Oriented Programming，OOP)，这就需要设计出能够支持面向对象程序设计方法的新语言。

Smalltalk 语言是最有影响力的面向对象语言之一，它丰富了面向对象的概念，包含许多面向对象的特征，例如类和继承等。在慢慢实践的过程中，人们发现由于 C 语言更加深入人心，使用广泛，以至于最好的方法不是去开发新的语言替代它，而是在其基础上进行扩展。在这种形势下，C++ 应运而生。C++ 是由美国 AT&T 公司 Bell(贝尔)实验室的 Bjarne Stroustrup(本贾尼·斯特劳斯特卢普)博士及其同事于 20 世纪 80 年代初在 C 语言的基础上开发成功的。C++ 保留了 C 语言原有的优点，增加了适用于面向对象程序设计的“类”(class)，因此最初命名为“带类的 C”，后来为了强调它是 C 语言的增强版，用了 C 语言中的自加运算符“++”，称为 C++，而 Bjarne Stroustrup 博士被尊称为 C++ 语言之父。

1.1.2 语言发展的进程

程序设计的任务就是用计算机懂得的语言(即程序设计语言)编写程序,然后交给计算机去执行。自从第一台计算机诞生以来,人类对计算机程序设计语言的研究一直没有间断过,程序设计语言的发展大致经过了以下三个阶段。

1. 机器语言

计算机的工作是基于二进制的,从根本上来说,计算机只能识别和接受由0和1组成的计算机指令。一条指令就是一个固定长度的由指令码和地址码组成的二进制位串,这就是计算机唯一可以读懂的语言,称为机器语言。程序员把要计算机完成的任务分解为一系列机器语言指令包括的动作,以指令序列的形式写出来,这就是机器语言程序设计。

这种语言运用简单,机器可以直接识别,但对于程序员来说却很不方便,完成一个简单的计算公式也要写几十条指令,编程工作枯燥,程序冗长,调试、修改、移植和维护都是难题,同时机器语言与人们的语言习惯差别太大,难学、难记、难修改、难检查、难写、难以推广和使用。

2. 汇编语言

为了克服机器语言的缺点,人们创造出了符号语言,它是用一些英文字母和数字来表示一个指令,例如:ADD代表"加",SUB代表"减"。但是计算机并不能直接识别和执行符号语言的指令,需要用一种称为汇编语言的软件,把符号语言的指令转换成机器语言。为此,符号语言又称为符号汇编语言或者是汇编语言。虽然汇编语言比机器语言较接近于人们的语言习惯,但是也很难普及,只在专业人员中使用,而且不同型号计算机的机器语言与汇编语言是互不通用的,机器语言和汇编语言是完全依赖于具体机器特性的,是面向机器的语言,称为计算机低级语言。

3. 高级语言

对于程序员来说,汇编语言比机器语言方便很多,但是仍然没有解决计算机编程难的基本问题,后来以FORTRAN和ALGOL 60为代表的高级语言逐渐流行,到了20世纪70年代,新一代的高级语言Pascal和C问世了。与汇编语言和机器语言相比,高级语言更接近人类的自然语言,当然计算机也不能直接识别高级语言编写的程序,要通过编译程序将高级语言编写的程序翻译成机器语言,再让计算机运行。

程序设计语言在经历了机器语言、汇编语言、高级语言后,进入了面向对象语言。面向对象的编程语言与以往各种编程语言的不同点在于:它设计的出发点就是为了能更直接地描述客观世界中存在的事物(即对象)以及它们之间的关系。

面向对象的编程语言可以直接描述问题域中的对象及其相互关系,它将客观事物看作具有属性和行为的对象,通过抽象找出同一类对象的共同属性(静态特征)和行为(动态

特征)并形成类,对象之间通过消息进行通信。面向对象语言实现了数据的封装,同时通过类的继承与多态实现了代码重用。

1.1.3　面向对象程序设计的基本概念

1. 类(Class)

类的概念来自于人们认识自然的过程,从一个个具体的事物中把共同的特征取出来,形成一个一般的概念,即为“类”。例如轿车、火车等可称为车类,蛇、虫等可称为爬行类,鱼、狗等可称为动物类,每种类中都具有一些共同的特性,如重量、长度等;共同的行为,如车类能行驶,爬行类能爬,动物类能动、叫等。类描述了一组具有相同特性(数据元素)和相同行为(函数)的对象。

2. 对象(Object)

同类的事物,分成许多具体的个体,把这些个体叫做“对象”。对象是现实世界中实际存在的事物,是类的一个具体实例,如某一辆具体的汽车、某一个具体的动物都是一个对象。

3. 属性(Attribute)

类中的特性(数据)称为类的属性,例如汽车的颜色、载重量等是汽车类的属性,动物的重量、高度等是动物类的属性。不同的类具有不同的属性。

4. 方法(Method)

类中的行为(函数)称为类的方法,如汽车类可以有加速方法、转向方法等,不同的类具有不同的方法。

C++ 是在 C 语言的基础上扩展而来的一门高效混合型实用程序设计语言,它包括两部分内容:一部分支持面向过程部分,是以 C 语言为核心的;另一部分支持面向对象,是 C++ 对 C 扩充的部分。

1.1.4　面向对象程序设计的特点

数据封装、继承和多态是面向对象程序设计的主要特征。

1. 数据封装

数据封装来源于现实世界,例如电视机发生故障需要更换一个晶体管,维修人员并不需要自己去做一个晶体管,而是在已经做好的晶体管中找到一个合适的使用,因此维修人员并不需要关心晶体管的内部结构,也没有必要了解其工作原理,只需要关心晶体管的各

种参数及其与外界的连接即可。

面向对象程序设计的一个重要特征就是数据封装，所谓数据封装是指将类的成员按其使用的存取方式进行分类，进而控制类成员的存取访问权限。面向对象的程序设计通过数据封装，使得用户对这些数据的使用必须通过适当的接口，否则无法使用某些数据，从而实现数据隐藏的目的。

2. 继承

当需要设计新型的汽车时，有两种方式可以选择：一种是从头开始设计；另一种是在已有车型的基础上进行改进，加入一些新的特性，称之为继承。面向对象程序设计通过继承实现了代码重用，即在已经存在类的基础上定义一个新的类，新类继承已有类的属性和方法，同时增加了自己的属性和方法。例如：人类具有姓名、性别、年龄等属性，学生除了具有人类的所有属性之外，还有学号、班级、成绩等属性。这样在定义了人类之后再定义学生类时，只需要让学生类从人类继承，不需要在学生类中重复声明姓名、性别、年龄等属性，实现了代码的重用。

3. 多态

多态(Polymorphism)按字面的意思就是“多种状态”。在面向对象语言中，接口的多种不同的实现方式即为多态，多态指同一个实体同时具有多种形式，它是面向对象程序设计(OOP)的一个重要特征。C++ 中的多态性具体体现在运行和编译两个方面。运行时多态是动态多态，其具体引用的对象在运行时才能确定。编译时多态是静态多态，在编译时就可以确定对象使用的形式。

1.1.5 C++ 程序设计语言的特点

C++ 程序设计是在 C 语言程序设计的基础上发展而来的，同时支持面向对象的程序设计，它主要有以下特点。

(1) C++ 继承了 C 语言的所有特点，包括语言简洁、紧凑、使用方便、灵活；拥有丰富的运算符；生成的目标代码质量高，程序执行效率高；可移植性好，等等。

(2) 增加了一些新的运算符，例如，::、new、delete 等，使得 C++ 应用起来更方便。

(3) 对 C 语言的某些方面进行了改进，如引入了 const 常量取代了 C 语言中的宏定义；引入了内联函数，提高了程序的效率；引入了引用的概念等。

(4) 允许函数重载，允许设置默认参数，提高了编程的灵活性，减少了冗余性。

(5) 支持面向过程和面向对象的方法。在 C++ 环境下既可以进行面向对象的程序设计，也可以进行面向过程的程序设计。

(6) 完全支持面向对象的数据封装、数据隐藏、继承和多态等特点。