

可下载教学资料

<http://www.tup.tsinghua.edu.cn>

21世纪普通高校计算机公共课程规划教材

# C++程序设计教程

苏成 主编

姜薇 孙仁科 副主编



清华大学出版社

21世纪普通高校计算机公共课程规划教材

# C++程序设计教程

苏成 主编

姜薇 孙仁科 副主编

清华大学出版社  
北京

## 内 容 简 介

本书是为以 C++ 语言作为程序设计入门语言的初学者而编写的,全书分为基础篇、能力篇和实验篇。基础篇共有 9 章,介绍 C++ 语言的语法、编程规范与技巧,有典型例题和大量习题。能力篇介绍 6 种常用算法的思想与趣味实例,这些贴近生活的实例可使学生触类旁通,举一反三。实验篇介绍了 14 个实验,每个实验由四部分组成,引导学生从分析程序、完善程序,到动手编程,最后得以进阶提高。

本书简洁易懂,深入浅出,内容取舍合理,重点突出,重视应用。针对初学者的思维特点和教材难点,每章末尾有常见错误分析板块,使学生少犯同样的错误。本书强调培养算法素养、良好编程风格以及面向对象的思维模式,例如第 10 章以“求三角形种类与面积”为例,讨论了一个小型的课程设计的开发过程,可以提高学生的综合编程能力。

本书是大学 C++ 程序设计教材,也可以作为程序设计爱好者自学以及参加计算机等级考试的参考资料。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

### 图书在版编目(CIP)数据

C++ 程序设计教程/苏成主编.--北京:清华大学出版社,2013.1

21 世纪普通高校计算机公共课程规划教材

ISBN 978-7-302-30516-3

I. ①C… II. ①苏… III. ①C 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2012)第 258068 号

责任编辑:魏江江 王冰飞

封面设计:傅瑞学

责任校对:焦丽丽

责任印制:何 芊

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](mailto:c-service@tup.tsinghua.edu.cn)

质量反馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者:北京鑫海金澳胶印有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:22.5 字 数:543 千字

版 次:2013 年 1 月第 1 版 印 次:2013 年 1 月第 1 次印刷

印 数:1~3000

定 价:34.50 元

产品编号:041774-01

# 出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程(简称‘质量工程’)”,通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

本系列教材立足于计算机公共课程领域,以公共基础课为主、专业基础课为辅,横向满足高校多层次教学的需要。在规划过程中体现了如下一些基本原则和特点。

(1) 面向多层次、多学科专业,强调计算机在各专业中的应用。教材内容坚持基本理论适度,反映各层次对基本理论和原理的需求,同时加强实践和应用环节。

(2) 反映教学需要,促进教学发展。教材要适应多样化的教学需要,正确把握教学内容和课程体系的改革方向,在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养,为学生知识、能力、素质协调发展创造条件。

(3) 实施精品战略,突出重点,保证质量。规划教材把重点放在公共基础课和专业基础课的教材建设上;特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现教学质量和教学改革成果的教材。

(4) 主张一纲多本,合理配套。基础课和专业基础课教材配套,同一门课程针对不同层次、面向不同专业的多本具有各自内容特点的教材。处理好教材统一性与多样化,基本教材与辅助教材、教学参考书,文字教材与软件教材的关系,实现教材系列资源配套。

(5) 依靠专家,择优选用。在制订教材规划时要依靠各课程专家在调查研究本课程教

材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主题。书稿完成后要认真实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平教材编写梯队才能保证教材的编写质量和建设力度,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪普通高校计算机公共课程规划教材编委会

联系人:魏江江 weijj@tup.tsinghua.edu.cn

# 前 言

---

C++语言是当今应用广泛的一种混合型的高级程序设计语言。它既保持了C语言的高效和精练,支持面向过程的程序设计的特点,同时又是面向对象语言的杰出代表。

C++语言是众多高级语言中比较难学的一种,一是因为内容庞大,规则繁多,使用灵活,既要讲面向过程的程序设计,又要讲面向对象的程序设计。一般讲授C++语言有两种方式,一种是先学C语言,再学C++语言;另一种是直接讲授C++语言。由于受学时限制,很多学校采用第二种方式。由于C++作为入门语言,没有其他高级语言基础,所以从初学者的思维角度,应该先快速地引导他们认识面向过程的程序设计,熟悉选择结构、循环结构、函数、数组、指针等概念,然后转向面向对象的程序设计,重点掌握封装、继承、多态的概念。对于C语言中的一些重要概念,比如二维数组、指针、结构体、共用体等,在C++中不进行深入讨论,淡化“指针”,突出“引用”,将“结构体”作为一种特殊的“类”处理。

在学习C++语言的过程中,会不可避免地遇到如何处理面向过程的程序设计与面向对象的程序设计的关系。笔者认为不应将二者对立起来,前者是掌握C++的基础,后者是思维模式的转变与提高。对于一些简单的问题,采用面向过程的方法自然、实用,而且面向过程的方法更容易突出算法思想;对于一些较大规模的问题,从模块化程序设计转变到类与对象是一种自然的过渡。本书第10章列举了一个“求三角形种类与面积”示例,用不同的方法解决一个问题,可以帮助学生理清程序设计思路,选择合适的方法逐步求精,进而完善程序功能。

## 本书的目标

通过本课程的学习,希望读者能掌握C++语言的基本规则和概念,具备编写和调试一些简单程序的能力,在解决实际问题过程中,能够有意识地运用基本算法,并建立面向对象的思维模式。最后,读者学习本课程也为通过C++二级考试打好基础。

## 本书的特色

(1) 详略得当,重点突出。本书以大学计算机基础教学的基本要求为依据,兼顾全国计算机等级考试(C++二级)大纲要求。C++语言的内容可以说是博大精深,如果把所有内容都罗列进来,不仅学时不够,而且学生也难以消化。所以,在内容选择上,一些不常用而且比较难的内容,比如异常处理、STL标准模板库等,都不在本书范围内。字符串处理是实际应用中经常遇到的问题,也是程序设计中的一个难点,关于这个难点,本书详细介绍了两种方法:传统的C语言处理方法和C++的string类的方法,读者可以体会各自的特点。

(2) 理论与实践并重,强调算法思想和编程能力的培养。本书分为基础篇、能力篇和实验篇三部分。基础篇为第1~9章,主要介绍C++语言的语法、规则以及基本的编程方法。能力篇为第10章,针对初学者普遍存在的“看得懂,不会编程序”的问题,首先介绍一些常用的算法及其应用,比如枚举法、递推法、迭代法、递归法、分治法、贪心法等,这些算法在不少

程序设计教材中也有涉及,但是都比较分散,本书集中讲解了各种算法,有利于读者系统地掌握算法的特点和应用技巧;然后讲解了开发一个综合实例的过程,从简单设计到逐步完善,从面向过程到面向对象,引导读者掌握开发一个有一定规模的程序的方法。实验篇包含14个实验,每个实验由分析程序、完善程序、编写程序和进阶提高四部分组成,遵循循序渐进、逐步提高的原则,先从验证性程序起步,然后是阅读和完善别人的程序,接着是学会独立编写和调试程序,最后是综合提高。

(3) 预先出错提醒,让学生尽量少犯错误,少走弯路。在第2~8章的每章最后,都有一节“常见错误分析”,总结初学者在编程中容易出现的错误以及容易混淆的概念。学生了解了编程中出错的位置和原因,才能不断进步,编写出正确的、高质量的程序。

(4) 遵循C++标准和规范。鉴于Visual C++ 6.0的广泛应用,并作为指定考试环境,本书所有程序都在该环境下调试通过。但是,程序要符合标准C++规范,对于Visual C++ 6.0不符合标准的地方以及一些bug给予明确说明,以保证程序在其他环境中也能正常运行。

(5) 习题丰富,类型多样。填空题包括概念填空题、完善程序题以及阅读程序、写运行结果等。编程题是初学者感觉比较困难的,需要加强训练。读者在学习有关内容和例题的基础上,可以编写出规范的、可读性好的程序,不局限于一个标准答案,鼓励“一题多解”,举一反三。简答题主要涉及一些初学者模糊不清的概念,通过这类题目可以使学生加深对基本概念的理解。

### 本书的内容

本书共分为3个部分。基础篇包括第1~9章的内容。第1章是C++语言概述;第2章是简单的程序设计,包括数据类型、变量、表达式以及选择结构与循环结构程序设计;第3章介绍用户自定义的数据类型,包括数组、指针、引用、字符串以及枚举、结构体与共用体等;第4章介绍函数以及程序结构、变量作用域与生存期等概念;第5章介绍类与对象;第6章介绍类的继承与派生;第7章介绍类的多态性;第8章介绍文件操作以及输入输出格式控制;第9章介绍函数模板和类模板。能力篇为第10章,首先介绍一些常用的算法,包括枚举法、递推法、迭代法、递归法、分治法、贪心法及模拟法等(动态规划、回溯法等超出本书范围,不作介绍);然后以“求三角形种类与面积”为例,讨论了编写一个有一定规模的程序的过程和方法。实验篇包含14个实验,每个实验由浅入深,包括4个组成部分,可以根据学时灵活裁剪。附录A包含一套C++笔试模拟试题和上机操作题,基本以历年C++二级考试真题为蓝本,可以参考检验读者掌握C++的程度。

本书是面向C++语言初学者的入门教材,也可作为计算机二级考试的参考书。

苏成编写第3~5、7~10章,姜薇编写第1~2章,孙仁科编写第5章;实验篇以及第2~10章的习题由苏成、陈廷杰编写;全书由苏成统稿。

在本书的编写过程中得到了计算机学院、教务处的支持与帮助,C++课程的有关任课教师提出了许多宝贵意见,杨文嘉老师提出了许多好的建议,在此一并致谢。最后,感谢清华大学出版社员工的辛勤劳动,并感谢魏江江老师的大力支持。

由于水平所限,书中难免存在疏漏之处,敬请广大读者批评指正。

编者

2012年10月

# 目 录

## 基础篇

第 1 章 C++语言概述 .....	3
1.1 C++语言的产生和发展 .....	3
1.2 C++语言的特点 .....	4
1.3 C++程序的结构 .....	4
1.3.1 简单的 C++程序实例 .....	4
1.3.2 C++程序结构分析 .....	5
1.4 C++程序的开发步骤和集成开发环境 .....	7
1.4.1 C++程序的开发步骤 .....	7
1.4.2 VC++ 6.0 集成开发环境 .....	8
习题 1 .....	16
第 2 章 简单的程序设计 .....	18
2.1 C++语言的字符集、标识符与关键字 .....	18
2.1.1 字符集 .....	18
2.1.2 标识符 .....	18
2.1.3 关键字 .....	18
2.2 数据类型 .....	19
2.2.1 C++语言的数据类型简介 .....	19
2.2.2 基本数据类型 .....	19
2.2.3 用 typedef 重定义类型 .....	20
2.3 常量与变量 .....	21
2.3.1 常量 .....	21
2.3.2 变量 .....	24
2.4 不同类型数据的转换 .....	26
2.4.1 隐式类型转换 .....	26
2.4.2 强制类型转换 .....	27
2.5 运算符与表达式 .....	27
2.5.1 C++运算符简介 .....	27

2.5.2	算术运算符与算术表达式 .....	28
2.5.3	赋值运算符与赋值表达式 .....	28
2.5.4	关系运算符与关系表达式 .....	29
2.5.5	逻辑运算符与逻辑表达式 .....	30
2.5.6	逗号运算符与逗号表达式 .....	30
2.5.7	位运算符 .....	31
2.6	C++语言的基本控制结构及语句 .....	32
2.6.1	C++语句概述 .....	32
2.6.2	顺序结构 .....	34
2.6.3	选择结构 .....	36
2.6.4	循环结构 .....	45
2.6.5	辅助控制语句 .....	51
2.7	常见错误分析 .....	53
	习题 2 .....	55
<b>第 3 章</b>	<b>构造数据类型 .....</b>	<b>62</b>
3.1	数组 .....	62
3.1.1	一维数组 .....	62
3.1.2	二维数组 .....	65
3.2	指针 .....	67
3.2.1	指针的概念 .....	67
3.2.2	指针的基本操作与运算 .....	69
3.2.3	指针与数组的关系 .....	71
3.2.4	动态内存分配 .....	73
3.2.5	用限定符 const 修饰指针 .....	76
3.3	引用 .....	77
3.3.1	引用的概念 .....	77
3.3.2	引用与指针的区别 .....	78
3.4	字符串 .....	78
3.4.1	字符数组 .....	79
3.4.2	字符指针 .....	82
3.4.3	C++风格的字符串处理方法——string 类 .....	83
3.5	枚举类型、结构体与共用体 .....	86
3.5.1	枚举类型 .....	86
3.5.2	结构体 .....	87
3.5.3	共用体 .....	89
3.6	常见错误分析 .....	90
	习题 3 .....	91

<b>第 4 章 函数</b> .....	95
4.1 函数的定义和调用 .....	96
4.1.1 函数的定义 .....	96
4.1.2 函数的调用 .....	97
4.1.3 函数的声明 .....	98
4.2 函数的参数传递 .....	99
4.2.1 形参与实参 .....	99
4.2.2 参数的传递 .....	99
4.2.3 数组和指针作为函数参数 .....	101
4.3 递归函数 .....	103
4.3.1 函数的嵌套调用 .....	103
4.3.2 函数的递归调用 .....	104
4.4 内联函数 .....	105
4.5 形参含有默认值 .....	106
4.6 函数重载 .....	106
4.6.1 函数重载的定义 .....	107
4.6.2 匹配函数重载的规则 .....	107
4.6.3 函数重载的注意事项 .....	108
4.7 系统函数 .....	108
4.8 变量的作用域与生存期 .....	111
4.8.1 作用域与可见域 .....	111
4.8.2 局部变量和全局变量 .....	112
4.8.3 变量的生存期与存储类型 .....	113
4.9 常见错误与典型示例 .....	115
4.9.1 常见错误分析 .....	115
4.9.2 典型示例 .....	116
习题 4 .....	118
<b>第 5 章 类与对象</b> .....	126
5.1 面向对象程序设计的基本概念 .....	126
5.1.1 抽象 .....	126
5.1.2 封装 .....	127
5.1.3 继承 .....	127
5.1.4 多态 .....	128
5.2 类与对象 .....	128
5.2.1 类的定义 .....	128
5.2.2 类成员的访问控制 .....	130
5.2.3 类的成员函数 .....	130

5.2.4	对象	132
5.3	构造函数与析构函数	134
5.3.1	构造函数	135
5.3.2	带默认参数的构造函数	137
5.3.3	复制构造函数	138
5.3.4	析构函数	140
5.4	对象数组和对象指针	141
5.4.1	对象数组	141
5.4.2	对象指针	143
5.4.3	this 指针	144
5.5	静态成员	145
5.5.1	静态数据成员	145
5.5.2	静态成员函数	147
5.6	友元	148
5.6.1	友元函数	148
5.6.2	友元类	149
5.7	常类型	150
5.7.1	常对象	151
5.7.2	用 const 修饰的类成员	151
5.8	常见错误与典型示例	153
	习题 5	157
<b>第 6 章</b>	<b>继承与派生</b>	<b>163</b>
6.1	继承与派生的概念	163
6.2	派生类	163
6.2.1	派生类的定义	163
6.2.2	派生类的成员组成	165
6.2.3	继承方式	166
6.3	派生类的构造函数和析构函数	170
6.3.1	派生类的构造函数	170
6.3.2	派生类的析构函数	174
6.4	多继承	175
6.4.1	多继承的定义	175
6.4.2	多继承引起的二义性问题	176
6.5	虚基类	178
6.5.1	虚基类的概念	178
6.5.2	虚基类及其派生类的构造函数	180
6.6	基类与派生类的赋值兼容	181

6.7 常见错误与典型示例 .....	183
习题 6 .....	185
<b>第 7 章 多态性</b> .....	<b>189</b>
7.1 多态性概述 .....	189
7.2 运算符重载 .....	190
7.2.1 运算符重载为成员函数 .....	191
7.2.2 运算符重载为非成员函数 .....	192
7.2.3 运算符重载的规则和限制 .....	193
7.2.4 运算符重载的应用 .....	193
7.3 虚函数 .....	195
7.4 纯虚函数和抽象类 .....	197
7.5 虚析构函数 .....	199
7.6 常见错误与典型示例 .....	200
习题 7 .....	201
<b>第 8 章 输入输出流</b> .....	<b>205</b>
8.1 流的概念 .....	205
8.2 输入输出重定向 .....	205
8.2.1 4 个标准的输入输出流对象 .....	205
8.2.2 输入重定向 .....	206
8.2.3 输出重定向 .....	207
8.3 输入输出格式控制 .....	209
8.4 文件操作 .....	210
8.4.1 C++ 文件概述 .....	210
8.4.2 文件的打开与关闭 .....	212
8.4.3 对文本文件的操作 .....	214
8.4.4 对二进制文件的操作 .....	217
8.5 常见错误分析 .....	223
习题 8 .....	223
<b>第 9 章 模板</b> .....	<b>226</b>
9.1 函数模板 .....	226
9.1.1 函数模板的定义 .....	226
9.1.2 函数模板的使用 .....	227
9.2 类模板 .....	230
9.1.1 类模板的定义 .....	230

9.1.2 类模板的使用	230
习题 9	232

## 能 力 篇

第 10 章 常用算法与综合实例	237
10.1 常用的算法	237
10.1.1 枚举法	237
10.1.2 递推法	241
10.1.3 递归法	244
10.1.4 递归与递推的比较	245
10.1.5 分治法	247
10.1.6 贪心法	249
10.1.7 模拟法	252
10.2 综合实例	253
10.2.1 用面向过程的方法求三角形种类和面积	253
10.2.2 用面向对象的方法求三角形种类和面积	256
习题 10	264

## 实 验 篇

实验 1 熟悉 C++ 开发环境和编程步骤	271
实验 2 数据类型与表达式	273
实验 3 条件与开关语句	275
实验 4 循环语句	278
实验 5 数组	281
实验 6 指针与字符串	285
实验 7 函数	289
实验 8 作用域、生存期、多文件结构	297
实验 9 类与对象	302
实验 10 继承与派生	307
实验 11 运算符重载	312
实验 12 多态性	316

实验 13 输入输出流 .....	319
实验 14 模板 .....	322
附录 A C++ 模拟试题 .....	326
附录 B ASCII 码字符表 .....	340
附录 C 常用的库函数 .....	342
参考文献 .....	344

# 基础篇

- 第1章 C++语言概述
- 第2章 简单的程序设计
- 第3章 构造数据类型
- 第4章 函数
- 第5章 类与对象
- 第6章 继承与派生
- 第7章 多态性
- 第8章 输入输出流
- 第9章 模板



### 1.1 C++语言的产生和发展

C++语言是20世纪80年代由美国AT&T贝尔实验室开发的通用程序设计语言,它是在C语言的基础上发展演化而来的,在程序设计的众多领域得到了广泛的应用。

C语言是1972年美国贝尔实验室的D. M. Ritchie在B语言基础上开发出来的程序设计语言,它最初用做UNIX操作系统的描述语言。后来,C语言又经过了多次改进。1977年出现了不依赖于具体机器的C语言版本。1978年以后,C语言被先后移植到各种机型的计算机上,并已独立于UNIX系统。由于C语言运算符和数据结构丰富,程序执行效率高,语言简洁灵活,可直接访问计算机的物理地址,兼有高级语言和低级语言的优点,有大量的库代码和较多的开发环境,具有良好的可读性和可移植性,支持结构化程序设计,所以很快广为流行,成为应用最广泛的程序设计语言之一。

然而,随着人们要求计算机解决问题的规模不断增大,C语言在处理大问题、复杂问题时产生了一些局限。如:C语言的类型检查机制相对较弱,使得程序中的一些错误不能在编译阶段由编译器检查出来;C语言本身几乎没有支持代码重用的语言结构;用C语言开发大型程序时,程序员就很难控制程序的复杂性;C语言不支持面向对象的程序设计。

为了克服C语言的上述局限,1980年美国贝尔实验室的Bjarne Stroustrup及其同事对C语言进行了扩充和改进,开发出了C++。它引入了类的机制,所以最初的C++被称为“带类的C”。在此后的数年,这个“带类的C”被用于十几个系统的开发,并取得了初步的成功。1983年正式取名为C++。1985年,由C++语言开发者编写的《C++程序设计语言》一书的出版宣布了C++ 1.0版的诞生。

C++语言设计的初衷是为了扩充C语言,并引入了面向对象的程序设计思想。同时,C++在设计时充分考虑了与C语言的兼容性,大量C语言的开发工作得以继承和发展,这是C++语言成功的重要原因之一。

C++和C的主要区别并非是对语法的扩充,而是其对数据抽象和面向对象程序设计方法的支持。C++语言开发的宗旨是使面向对象设计技术和数据抽象成为软件开发的一种真正的实用技术。因此,C++语言的形成是一个不断发展和完善的过程,其研制和开发的过程以编译系统的有效实现为前提,这也是C++语言能够成功的重要原因。

自C++ 1.0推出以后的多年中,它的基本概念仍然在不断地扩充和形成。Bjarne Stroustrup和他的同事们又为C++引进了运算符重载、引用、虚函数等许多特性,使之更加