



普通高等教育“十一五”国家级规划教材

王志新 王大伦 编著

C/C++ 程序设计 上机指导

21 世纪
计 算 机 科 学 与 技 术
实 践 型 教 程

丛书主编
陈明



清华大学出版社

TP312/2389C

2008

王志新 王大伦 编著

C/C++ 程序设计 上机指导

21 世纪计算机科学与技术实践型教程

丛书主编 陈明

清华大学出版社
北京

内 容 简 介

本书配合教材《C/C++ 程序设计实用教程》(王大伦编著,清华大学出版社),为读者提供上机服务。本书将程序设计分为 56 个题目进行讨论。每个题目给出典型的演示程序,并提供了点评。

由于本书所附程序面广量多,有利于因材施教,故适用于高等院校和职业院校学生上机、自学之用。本书也可作为“C/C++ 程序设计”课程的教学参考书,为任课教师提供实验课教学参考。

全书的源程序以及《C/C++ 程序设计实用教程》一书中的所有讲座和载有 600 多个程序的 P、L、R 目录均可下载使用。作者研制了几个未见于其他教材的实用类:两种坐标系统的复数类、矩阵类、多项式类,并说明怎样借助这几个实用类,用 C++ 程序设计技巧解决数字滤波问题。这些实用类均可下载使用。全部程序都在 Visual C++ 6.0 环境下核实无误。

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/C++ 程序设计上机指导/王志新,王大伦编著. —北京:清华大学出版社,2008.5
(21 世纪计算机科学与技术实践型教程)
ISBN 978-7-302-16868-3

I. C… II. ①王… ②王… III. C 语言—程序设计—高等学校—教学参考资料
IV. TP312

中国版本图书馆 CIP 数据核字(2008)第 006799 号

责任编辑:谢琛 李晔

责任校对:时翠兰

责任印制:何 芊

出版发行:清华大学出版社

<http://www.tup.com.cn>

社 总 机:010-62770175

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

地 址:北京清华大学学研大厦 A 座

邮 编:100084

邮 购:010-62786544

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:185×260 印 张:18.5

字 数:421 千字

版 次:2008 年 5 月第 1 版

印 次:2008 年 5 月第 1 次印刷

印 数:1~4000

定 价:27.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:010-62770177 转 3103 产品编号:027651-01

《21 世纪计算机科学与技术实践型教程》

序

21 世纪影响世界的三大关键技术是：以计算机和网络为代表的信息技术；以基因工程为代表的生命科学和生物技术；以纳米技术为代表的新型材料技术。信息技术居三大关键技术之首。国民经济的发展采取信息化带动现代化的方针，要求在所有领域中迅速推广信息技术，导致需要大量的计算机科学与技术领域的优秀人才。

计算机科学与技术的广泛应用是计算机学科发展的原动力，计算机科学是一门应用科学。因此，计算机学科的优秀人才不仅应具有坚实的科学理论基础，而且更重要的是能将理论与实践相结合，并具有解决实际问题的能力。培养计算机科学与技术的优秀人才是社会的需要、国民经济发展的需要。

制定科学的教学计划对于培养计算机科学与技术人才十分重要，而教材的选择是实施教学计划的一个重要组成部分，《21 世纪计算机科学与技术实践型教程》主要考虑了下述两方面。

一方面，高等学校的计算机科学与技术专业的学生，在学习了基本的必修课和部分选修课程之后，立刻进行计算机应用系统的软件和硬件开发与应用尚存在一些困难，而《21 世纪计算机科学与技术实践型教程》就是为了填补这部分鸿沟。将理论与实际联系起来，结合起来，使学生不仅学会了计算机科学理论，而且也学会应用这些理论解决实际问题。

另一方面，计算机科学与技术专业的课程内容需要经过实践练习，才能深刻理解和掌握。因此，本套教材增强了实践性、应用性和可理解性，并在体例上做了改进——使用案例说明。

实践型教学占有重要的位置，不仅体现了理论和实践紧密结合的学科特征，而且对于提高学生的综合素质，培养学生的创新精神与实践能力有特殊的作用。因此，研究和撰写实践型教材是必需的，也是十分重要的任务。优秀的教材是保证高水平教学的重要因素，选择水平高、内容新、实践性强的教材可以促进课堂教学质量的快速提升。在教学中，应用实践型教材可以增强学生的认知能力、创新能力、实践能力以及团队协作和交流表达能力。

实践型教材应由教学经验丰富、实际应用经验丰富的教师撰写。此系列教材的作者，不但从事多年的计算机教学，而且参加并完成了多项计算机类的科研项目，把他们积累的经验、知识、智慧、素质融合于教材中，奉献给计算机科学与技术的教学。

我们在组织本系列教材过程中，虽然经过了详细的思考和讨论，但毕竟是初步的尝试，不完善甚至缺陷不可避免，敬请读者指正。

本系列教材主编 陈明

2005 年 1 月于北京

前 言

“学而时习之，不亦乐乎。”孔子这句常学常新、深含哲理的名言对于真正学好 C/C++ 编程，也具有指导意义。我们认为，这句话中的“习”并不只是看书、复习，而是包括演练、上机操作。对 C/C++ 程序设计教学来说，这句话的确切意思是：学了 C/C++ 的语句和编程理论，课后照例复习一遍，是远远不够的；在学习过程中还要不断实践。只有培养出追求真理的兴趣，才能最终对实际问题应付自如，即所谓从“必然王国”进入“自由王国”。

我们认为，要“时习”的并不只是教材本身及一般教材所指定的少数几个“实验”。实践证明，就算把教材读了多遍，完成了教材指定的那几个“实验”，遇到具体编程问题时，还是可能束手无策。我们所要培养的创新人才应能独立思考，并初步具有解决实际问题的能力。本书试图朝着这个方向进行探索。

由于各院校的具体情况很不相同，本书并不是作为一般意义上的“实验指示书”，简单地给出几个实验程序，让学生照着上机，加以验证。考虑到不同层次的读者有不同的要求，为了利于教师因材施教，在出版《C/C++ 程序设计实用教程》(清华大学出版社，2006 年)一书(以下简称“教材”)时，曾给出录入到 P、L、R 三个目录的大量 C/C++ 程序和多个讲座文稿，读者可以从 <http://www.tup.tsinghua.edu.cn>^① 下载。其中，P 目录给出书中正文涉及的程序；L 目录给出讲座所用的程序；R 目录除收集以上两个目录的程序，还添加了许多例子。本书作为 C/C++ 程序设计上机指导，将 C/C++ 程序设计问题分成 56 个课题加以阐述。书中的部分程序选自上述几个目录和文稿，并多有增补。在 myC_CPP 目录中保存有书中的所有程序，同时也包含上述 P、L、R 三个目录和多个讲座文稿。几百个程序是让读者非常容易上机的巨大资源。我们用 Visual C++ 6.0 编译系统核查了所有程序，并给出了比较详细的注解和部分运行结果；对错误语句，也指出了错误原因。为了利于读者分析书中所给程序，书中并未列出所有程序的运行结果。读者可预测运行结果，然后与网站中所给的“谜底”相比较。

总的来说，所选的 56 个课题不仅会帮助读者巩固书本知识，更重要的是，将会帮助读者掌握设计高质量程序的能力。我们将用实例给读者带来一般教科书少有的新意。例如，这些实例将说明：

- 怎样正确地实施多文件的编译与连接。

^① 进入 <http://www.tup.tsinghua.edu.cn> 后用书名搜索即可找到本书相关资源。如下载不便，可向本社来函、来电索取。

- 怎样通过多文件编程实现信息隐蔽。
- 怎样调试程序。
- 怎样分析程序设计中的难点(函数设计、访问二维数组的几种方法、用指针访问中文字符数组等)。
- 常见的编程方法有哪几种。
- 怎样设计 C++ 的实用类(如基于直角坐标和极坐标的复数类、矩阵类、多项式类等)以及如何用于实践中(例如,计算 FIR 数字滤波器的频率特性等)。

在使用本书前,请将 myC_CPP 目录复制于 C 盘,存放上机程序和有关文件。为了提高学习效率,建议读者:

- 在上机前,仔细阅读课题介绍以及教材(《C/C++ 程序设计实用教程》)的有关章节。要彻底弄懂基本语句的格式。
- 在上机前,逐行逐句读懂所给程序,预测运行结果。
- 理解并执行书中程序后,请合上书本,自行编制该程序。
- 勤于上机,勤于思考。大胆修改程序。大胆怀疑程序是否有错,是否合理,是否能采用另一种实现方案。

本书分两部分。第一部分介绍 C 语言编程,第二部分介绍 C++ 语言编程。这两部分的程序各用字母 A 和 B 为前缀。例如: A0202. CPP 表明该文件是 C 语言源文件,是课题 A02 的第二个程序的主文件; A0202A. h 是头文件, A0202A. CPP 是与之相应的实现文件。课题 A02 的所有程序文件都存入目录 C:\myC_CPP\A02 中。每个课题可能有多个程序,但每个程序只有一个主文件。

作者总结多年教学经验写成本书,但错误和不足仍在所难免,恳请读者不吝指出。

我们的邮箱: dalunwang@126.com。

王志新 王大伦

2008 年 2 月

目 录

第一部分 C 语言程序	1
A01 创建 Hello World 程序(简单的单文件程序)	1
A01.1 创建项目	1
A01.2 建立文件	3
A01.3 编译	4
A01.4 连接与运行	4
A01.5 复制程序运行结果	5
A01.6 Workspace 文件	5
A01.7 建立程序文件的简易方法	6
A01.8 注意事项	6
实验 A01	6
A02 源代码隐蔽机制(多文件程序的编译与连接)	7
实验 A02	10
A03 运算符 sizeof 的用法和几种类型的数据所占的字节数	10
实验 A03	11
A04 printf 函数和 scanf 函数中的基本格式字符的用法	11
A04.1 printf()函数中的基本格式字符	12
A04.2 scanf 函数中的基本格式字符	14
实验 A04	17
A05 初识 C++ 的 I/O 流	18
实验 A05	20
A06 调试程序	20
A06.1 VC++ 的程序调试功能	20
A06.2 检查数据	24
A06.3 复制程序运行结果	25
实验 A06	25
A07 选择语句——if 语句和 switch 语句	25
A07.1 条件语句(if 语句)	25

A07.2	条件语句的嵌套	27
A07.3	多分支条件语句的困惑	28
A07.4	开关语句——switch 语句	28
实验 A07		30
A08	循环语句 for——迭代法和穷举法	30
A08.1	迭代法	31
A08.2	continue 语句和 break 语句的用法和区别	32
A08.3	穷举法	33
A08.4	for 语句的变异	34
实验 A08		35
A09	循环语句 do...while 和 while——迭代法	35
实验 A09		36
A10	函数初步	37
A10.1	函数格式	37
A10.2	栈操作	37
A10.3	函数原型	39
A10.4	参数传递	39
实验 A10		42
A11	重载函数与具有默认参数的函数	42
A11.1	重载函数	42
A11.2	具有默认参数的函数	44
实验 A11		45
A12	递归函数	45
实验 A12		47
A13	程序结构	48
A13.1	单文件中的全局变量和局部变量	48
A13.2	单文件中的静态变量	50
A13.3	多文件中的变量与函数	51
实验 A13		53
A14	条件编译指令的应用	54
实验 A14		55
A15	数组与指针	56
A15.1	数组的基本知识	56
A15.2	指针的基本知识	57
A15.3	访问一维数组的几种方法	60
A15.4	二维数组元素在内存中的排列——降维处理	61
A15.5	访问二维数组的几种方法	62
A15.6	探索法——设计魔方图	64

实验 A15	64
A16 数组与指针的应用	65
实验 A16	69
A17 对指针及其指向对象的修饰	69
A17.1 常量指针(const 对象)	69
A17.2 指针常量(const 指针)	71
A17.3 指向常量的指针常量	72
实验 A17	73
A18 堆内存分配——采样数据处理	74
A18.1 函数 malloc() 和 free()	74
A18.2 运算符 new 和 delete	75
实验 A18	78
A19 引用、函数参数表与返回量(一)	78
A19.1 引用	78
A19.2 函数参数表	81
A19.3 返回量	85
实验 A19	86
A20 引用、函数参数表与返回量(二)	87
实验 A20	89
A21 字符和字符串处理函数	89
A21.1 字符串长度	90
A21.2 字符串复制	90
A21.3 字符串比较	90
A21.4 字符串串接	91
实验 A21	92
A22 函数指针	92
A22.1 函数指针的定义	92
A22.2 用函数指针变量调用函数	93
A22.3 函数指针数组	94
A22.4 用指向函数的指针作函数参数	96
实验 A22	97
A23 枚举类型、结构体和共同体	97
A23.1 枚举类型	97
A23.2 结构体	100
A23.3 共同体	104
实验 A23	107
A24 链表	107
A24.1 链表的概念	107

A24.2	结点的构成与访问	108
A24.3	链表的两种构造方法	109
实验 A24		110
A25	简单链表操作	111
实验 A25		114
A26	C 文件的读写	114
A26.1	文件类型指针	114
A26.2	文件的打开与关闭	115
A26.3	顺序文件的读写操作——函数 fprintf()、fscanf() 和 feof()	116
A26.4	几个读写函数——fputs()、fgets()、putc() 和 fgetc() 的用法	117
实验 A26		120
第二部分	C++ 语言程序	121
B01	认识类与对象	121
B01.1	类的组成	121
B01.2	内联函数	122
B01.3	对象的定义	123
B01.4	访问类的成员	123
实验 B01		125
B02	初识构造函数和析构函数	125
B02.1	构造函数的作用与特点	125
B02.2	构造函数的调用次序	127
B02.3	构造函数的初始化列表	128
B02.4	析构函数	129
实验 B02		130
B03	对象在内存中的布局 and this 指针	130
实验 B03		132
B04	深入认识构造函数	132
B04.1	默认构造函数	132
B04.2	复制构造函数	133
B04.3	具有默认参数的构造函数	136
B04.4	重载构造函数	137
实验 B04		138
B05	堆对象	138
B05.1	创建和删除单个堆对象的方法	139
B05.2	创建和删除堆对象数组的方法	140
实验 B05		141
B06	静态数据成员和静态成员函数	141

B06.1	静态数据成员的特点及定义方法	141
B06.2	访问静态数据成员的方法	142
B06.3	静态成员函数	142
实验 B06		145
B07	无名对象与临时对象	145
实验 B07		146
B08	常量对象和常量成员函数	146
实验 B08		148
B09	类成员	148
实验 B09		150
B10	友元	150
B10.1	为什么需要友元	150
B10.2	友元的定义方法	150
B10.3	友元函数的特点	151
实验 B10		154
B11	公有、保护和私有继承	154
B11.1	继承的概念	154
B11.2	公有、保护和私有继承	155
实验 B11		158
B12	类层次中的构造函数和析构函数的调用	159
B12.1	构造函数的调用次序	159
B12.2	析构函数的调用次序	159
实验 B12		161
B13	派生类的初始化方法——初始化列表的用法	161
实验 B13		163
B14	重载和操纵	164
实验 B14		165
B15	类层次中的对象数据结构和指针应用	166
实验 B15		169
B16	多继承与虚函数	169
B16.1	没有公共基类的多继承——简单的多继承	169
B16.2	具有公共基类的多继承	171
实验 B16		175
B17	虚基类	176
实验 B17		180
B18	多态性与虚函数	180
B18.1	多态性	180
B18.2	联编	180

B18.3	静态多态性	181
B18.4	虚函数和动态多态性	182
B18.5	对虚函数的限制	184
	实验 B18	186
B19	纯虚函数和抽象类	186
	实验 B19	191
B20	运算符重载(一)——四则运算符的重载	191
	B20.1 运算符重载基础	191
	B20.2 运算符重载函数	194
	实验 B20	196
B21	运算符重载(二)——赋值运算符以及自增 1、自减 1 和 下标运算符的重载	197
	B21.1 赋值运算符的重载	197
	B21.2 自增 1 和自减 1 运算符的重载	197
	B21.3 下标运算符的重载	199
	实验 B21	202
B22	运算符重载(三)——关系运算符以及输入输出运算符的重载	202
	B22.1 关系运算符的重载	202
	B22.2 组合运算符的重载	203
	B22.3 输出与输入运算符的重载	205
	实验 B22	207
B23	类型转换	207
	B23.1 基本类型转换为自定义类型	207
	B23.2 自定义类型转换为基本类型	209
	B23.3 自定义类型之间的转换	210
	实验 B23	213
B24	I/O 流(一)——访问顺序文件和二进制文件	214
	B24.1 I/O 流的概念	214
	B24.2 文件与文件流概述	214
	B24.3 文件的打开和关闭操作	215
	B24.4 顺序文件的读写操作	217
	B24.5 二进制文件的读写操作	220
	实验 B24	223
B25	I/O 流(二)——访问随机文件	223
	B25.1 随机文件的特点	223
	B25.2 随机文件的读、写方法	224
	实验 B25	227
B26	模板	227

B26.1	模板的概念	228
B26.2	函数模板与模板函数	229
B26.3	类模板与模板类	232
实验 B26		234
B27	复数类	236
实验 B27		244
B28	矩阵类	245
实验 B28		256
B29	多项式类	256
实验 B29		265
B30	C++ 在数字滤波器设计中的应用	265
实验 B30		268
附录 A	与字符和字符串有关的库函数	269
附录 B	ASCII 码表简介	272
附录 C	网站资源目录	276

第一部分 C 语言程序

A01 创建 Hello World 程序(简单的单文件程序)

本书中的程序并不是带有控件的 Windows 应用程序,而是所谓“工作台应用程序”(Console Application)。程序的创建过程包括程序文件(C / C++ 源文件、头文件)的编写、编译、连接和调试。整个过程在常用的 Visual C++ 6.0(简称 VC++)系统的 IDE(集成开发环境)中进行。

许多程序设计教科书都喜欢用 Hello World 程序与读者行见面礼。这里,我们照例首先创建这样一个非常简单的单文件程序,它只在屏幕上显示字符串“Hello World”。为此,要首先建立一个项目,然后在其中建立 C / C++ 源文件。

A01.1 创建项目

安装了 Visual C++ 6.0 之后,双击桌面上的 VC 快捷图标,便可进入 VC++ 集成开发环境,显示 VC++ 的 IDE 主界面,如图 A01-1 所示。

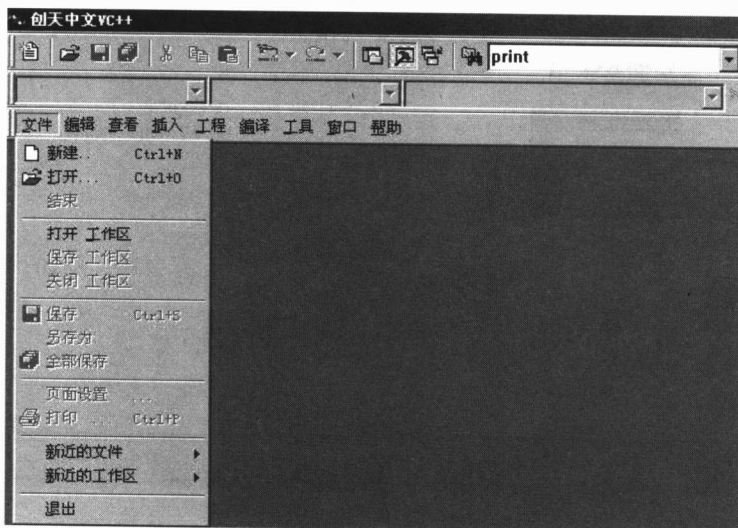


图 A01-1 VC++ 的 IDE 主界面

在菜单栏的“文件”菜单(如图 A01-1 所示)下,单击“新建”命令。在出现的对话框中,选中“工程”选项卡,如图 A01-2 所示。在该对话框的左边,选中 Win32 Console Application(32 位控制台应用程序)选项,而在右边填入路径(例如,C:\myC_CPP\A01)和项目名(例如,A0101)后,单击“确定”按钮。出现如图 A01-3 所示的对话框,询问要建立哪种控制台应用程序。这时,选中 An empty project 单选钮,单击“完成”按钮。其后,出现另一个信息框(如图 A01-4 所示)。确认后,就以 C:\myC_CPP\A01 为路径,建立了一个名为 A0101 的空项目(空目录)。以后有关该项目的文件,例如,项目文件、源文件、头文件等均被保存于此。

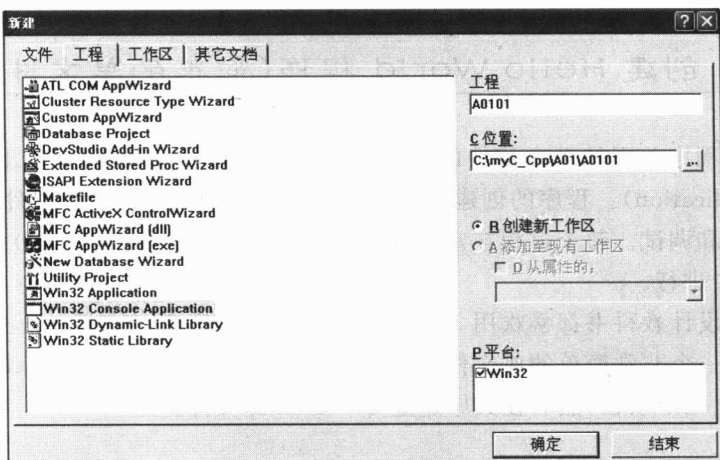


图 A01-2 建立项目

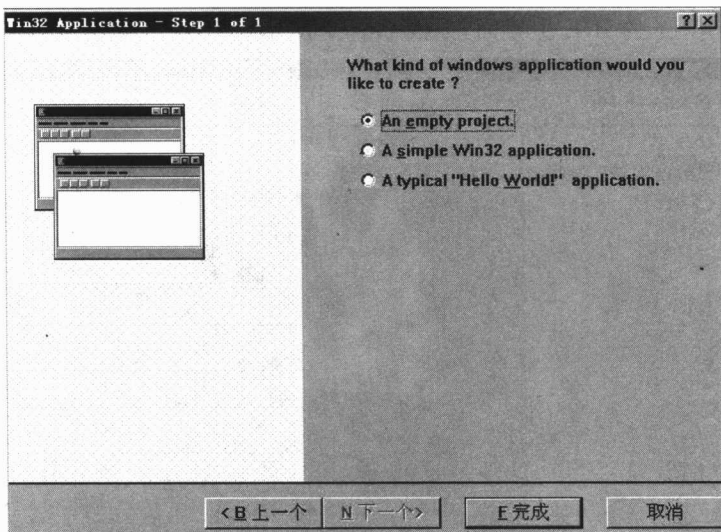


图 A01-3 询问框

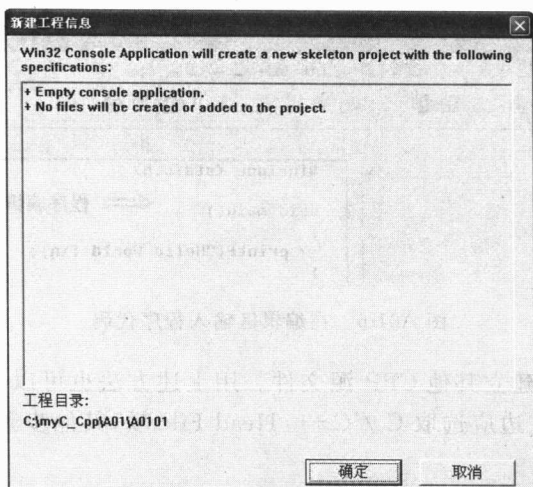


图 A01-4 信息框

A01.2 建立文件

按上述方法建立项目后,接着要在这个项目中建立 CPP 源文件。

在 IDE 中,单击“文件”|“新建”命令,便出现如图 A01-5 所示的对话框。如果要建立 CPP 源文件,应选中对话框中左边的 C++ Source File 选项。对话框右边的第一栏是已建立的项目名称,第三栏是项目的路径。在第二栏输入文件名,例如,A0101(默认扩展名是.CPP)。确认后,便在项目中建立了空的源文件 A0101.CPP,并显示如图 A01-6 所示的界面。图 A01-6 中右边的区域出现闪烁的光标,这是程序编辑区。在该区域输入代码,得到源程序文件(A0101.CPP)。

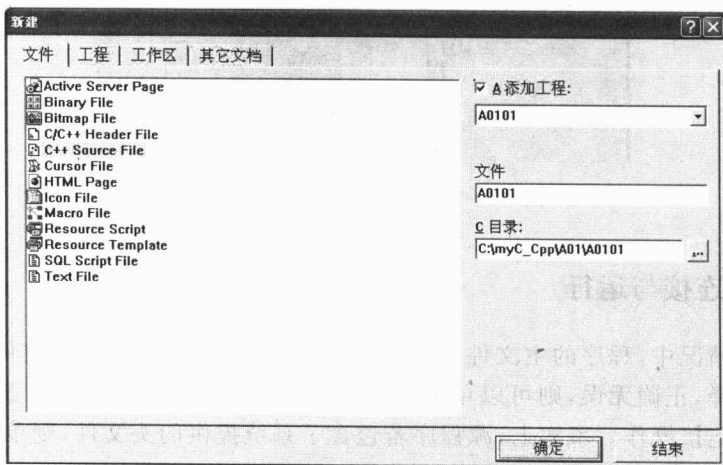


图 A01-5 “新建”对话框

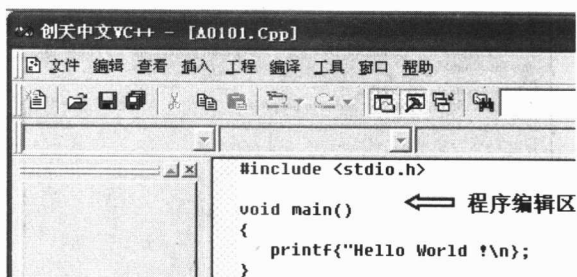


图 A01-6 在编辑区输入程序代码

重复上述操作,可建立其他 CPP 源文件。用上述方法也可以建立头文件,即在如图 A01-5 所示的对话框左边应选取 C / C++ Head File 项,其他步骤同上。头文件的默认扩展名是 .H。

完成上述操作后,在指定的路径上生成一个与工程名字相同的目录,CPP 源文件位于其中。当然,CPP 源文件名字也可以有别于工程名字。

⚠ 注意: 所有 CPP 源文件中,只能有一个文件带有主函数 main()。

A01.3 编译

如果程序编辑区已出现待编译的主文件(包含函数 main() 的源文件),可用“编译”命令(如图 A01-7 所示)对其进行编译。否则,单击“文件”|“打开”命令,输入文件名,使待编译的主文件出现在编辑区中,然后进行编译。

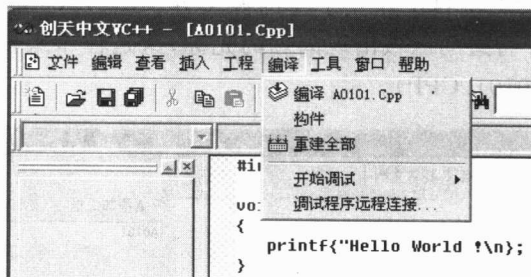


图 A01-7 用“编译”命令编译文件

A01.4 连接与运行

在单文件情况中,程序的主文件没有包含自定义的 CPP 文件或头文件。这时,如果主文件经过编译,正确无误,则可以单击“执行”命令(如图 A01-8 所示),执行该程序,不必显式地进行连接操作。事实上,源程序若包含了系统提供的头文件,则单击“执行”命令后,系统就自动将该头文件(它只是一张“菜单”,其中罗列了可供用户使用的一些函数)的实现文件(CPP 源文件)与主文件进行连接,如果正确,就生成可供运行的执行代码。一个项目只能有一个主文件。