



高职高专教育软件技术专业核心课程系列教材
国家骨干高等职业院校建设项目成果

C++应用程序设计

主 编 刘彦姝 谢景伟
副主编 戴 臻 胡 君 杨 夏 戴霞飞

C++YINGYONGCHENGXUSHEJI



中国水利水电出版社
www.waterpub.com.cn

高职高专教育软件技术专业核心课程系列教材

C++应用程序设计

主 编 刘彦姝 谢景伟

副主编 戴 臻 胡 君 杨 夏 戴霞飞



中国水利水电出版社

www.waterpub.com.cn

内容提要·前言·目录

内 容 提 要

本书是国家骨干高等职业院校建设项目成果。本书通过大量项目案例,主要介绍了C++基础、运算符及结构化语句、结构体及数组、指针及其应用、函数、算法基础等知识。全书遵循学习者的认知规律,秉承“任务驱动教学法”和“案例式教学法”,采用教学做合一、学以致用的编写思路,力求帮助读者在学习理解与积累知识,在应用的基础上创造性地实践,最终掌握C++程序的开发技术。

本教材可作为应用型、技能型人才培养的软件技术相关专业的“C++程序设计”课程的教学用书,也可作为各类培训、计算机从业人员和计算机爱好者的参考用书。

本书提供源代码,读者可以从中国水利水电出版社网站以及万水书苑下载,网址为:
<http://www.waterpub.com.cn/softdown> 或 <http://www.wsbookshow.com/>。

图书在版编目(CIP)数据

C++应用程序设计 / 刘彦姝, 谢景伟主编. — 北京:
中国水利水电出版社, 2013.8
高职高专教育软件技术专业核心课程系列教材
ISBN 978-7-5170-1103-3

I. ①C… II. ①刘… ②谢… III. ①
C程序—程序设计—高等职业教育—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2013)第172913号

策划编辑: 雷顺加 责任编辑: 李 炎 加工编辑: 孙 丹 封面设计: 李 佳

书 名	高职高专教育软件技术专业核心课程系列教材 C++应用程序设计
作 者	主 编 刘彦姝 谢景伟 副主编 戴 臻 胡 君 杨 夏 戴霞飞
出版发行	中国水利水电出版社 (北京市海淀区玉渊潭南路1号D座 100038) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn
经 售	电话: (010) 68367658 (发行部)、82562819 (万水) 北京科水图书销售中心(零售) 电话: (010) 88383994、63202643、68545874 全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	三河市鑫金马印装有限公司
规 格	184mm×260mm 16开本 13.25印张 330千字
版 次	2013年8月第1版 2013年8月第1次印刷
印 数	0001—3000册
定 价	26.00元

凡购买我社图书,如有缺页、倒页、脱页的,本社发行部负责调换
版权所有·侵权必究

前 言

C++是近年来国内外广泛使用的现代计算机语言，它既支持基于过程的程序设计，也支持面向对象的程序设计。C++继承了C语言的编程风格，不仅语法严谨，而且功能强大、内容丰富，因而成为编程人员使用最广泛的工具。国内许多高校陆续开设了“C++程序设计”课程，学好C++可以很容易触类旁通其他软件，C++架起了通向强大、易用、真正的软件开发应用的桥梁。

本书以项目化、工作过程化的方式深入浅出地介绍了程序开发的方法，不再拘泥于语法细节，重点通过使用语言来编写程序，使读者从实践中学习并巩固相关基本知识。

本书的特点：

- 从C++的特色写起，以类型定义、识别为安全中心，注重编程质量；
- 从自身学习经历写起，将经验与体会作为链接知识与能力的纽带；
- 从初学角度写起，点透要害，巧做取舍；
- 代码风格独特，实验上手较快，不需要复杂的调试。

本书每章都设置了工作情境、解决方案和能力目标，并从每个目标下分解出若干个任务，在每个任务中又按“任务分析”、“相关知识”、“实现方法”来展开，最后进行归纳小结，并给出适当的作业进行补充。这种方式可以使读者在明确要完成的任务的前提下学习知识，边学边做，训练技能，同时配有练习，能进一步锻炼和培养读者编程能力。

本书认真分析了读者在学习中的困难和认知规律，降低了入门起点，不需要C语言的基础，从零开始介绍程序设计和C++语言，可以作为高职高专计算机相关专业的教材，建议学时安排在70学时左右，其中讲授和上机实践学时比例在1:1左右。

本书由刘彦姝、谢景伟任主编，戴臻、胡君、杨夏、戴霞飞任副主编。本书在编写过程中得到了许多职业院校老师和企业工程师的大力支持，参考了一些作者的相关著作和文献，在此一并表示感谢。由于作者水平有限，书中难免存在不少缺点和错误，希望广大读者批评指正。我们的邮箱是：CProject@163.com。

编 者
2013年6月

目 录

前言

第1章 C++基础.....1

任务 1.1 C++入门.....1

1.1.1 程序和程序语言.....2

1.1.2 C++程序结构.....3

1.1.3 C++程序的书写格式和风格.....5

1.1.4 C++程序设计基本流程.....6

1.1.5 C++开发环境简介.....7

任务 1.2 数据类型与表达式.....15

1.2.1 基本数据类型.....16

1.2.2 常量.....17

1.2.3 变量.....20

1.2.4 运算符.....22

1.2.5 表达式与类型转换.....28

任务 1.3 基本输入/输出.....30

1.3.1 编译预处理.....30

1.3.2 标准输入/输出函数.....31

1.3.3 字符输入/输出函数.....32

1.3.4 格式化输入/输出函数.....33

第2章 运算符及结构化语句.....38

任务 2.1 运算符与表达式.....38

2.1.1 赋值运算符.....39

2.1.2 算术运算符.....39

2.1.3 算术类型转换.....41

2.1.4 关系和逻辑运算符.....42

2.1.5 条件运算符.....42

2.1.6 位运算符.....43

2.1.7 复合赋值运算符.....44

2.1.8 表达式.....45

任务 2.2 结构化语句.....50

2.2.1 顺序结构.....50

2.2.2 if 语句.....51

2.2.3 switch 语句.....57

2.2.4 while 语句.....63

2.2.5 do...while 语句.....65

2.2.6 for 语句.....68

2.2.7 break 语句.....70

2.2.8 continue 语句.....72

2.2.9 循环嵌套.....73

第3章 结构体及数组.....86

任务 3.1 结构体.....86

3.1.1 结构体类型的定义.....87

3.1.2 结构体变量的定义.....88

3.1.3 结构体变量的初始化.....90

3.1.4 结构体变量的引用.....91

任务 3.2 数组.....97

3.2.1 一维数组的定义.....97

3.2.2 一维数组的初始化.....98

3.2.3 一维数组元素的访问.....99

3.2.4 一维数组的应用.....101

3.2.5 二维数组的定义.....103

3.2.6 二维数组的初始化.....103

3.2.7 二维数组元素的访问.....104

3.2.8 二维数组的应用.....106

3.2.9 数组的应用.....108

第4章 指针及其应用.....122

任务 4.1 指针.....122

4.1.1 指针的概念.....123

4.1.2 指针变量的定义.....124

4.1.3 引用指针变量.....126

任务 4.2 指针和数组.....134

4.2.1 指向数组元素的指针.....134

4.2.2 字符串与指针.....141

任务 4.3 指针的应用.....150

4.3.1 指针移动.....150

4.3.2 指针作为函数参数.....153

第5章 函数.....162

任务 5.1 函数基础知识.....162

5.1.1 函数的定义.....163

5.1.2 函数的声明	165	6.2.1 枚举算法的基本概念	188
5.1.3 函数的参数	166	6.2.2 枚举算法的应用	188
5.1.4 函数的调用	171	任务 6.3 迭代法	192
5.1.5 常见的字符串库函数	173	6.3.1 迭代算法的基本概念	193
任务 5.2 变量的作用域与存储类型	177	6.3.2 迭代算法的应用	193
5.2.1 变量的作用域	178	任务 6.4 递归法	196
5.2.2 变量的存储类型	181	6.4.1 递归算法的基本概念	196
第 6 章 算法基础	184	6.4.2 递归算法的应用	198
任务 6.1 算法的概念	184	任务 6.5 排序	200
6.1.1 算法的定义	185	6.5.1 排序的基本概念	201
6.1.2 算法的特征	185	6.5.2 直接插入排序	201
6.1.3 算法评价	185	6.5.3 冒泡排序	203
任务 6.2 枚举法	187	参考文献	206

C++基础

工作情境

随着公司业务不断发展，各类员工越来越多，为了降低财务管理人员的工作强度，保证财务计算的准确性，公司需要一套完整的工资系统来对员工的工资进行管理，实现对各级员工工资的录入、计算和查询等操作，先决定指派你来完成工资管理模块的设计和实现。

解决方案

要实现对员工工资的管理，财务人员需要通过从键盘等输入设备上输入员工的档案工资、出勤情况、绩效工资等信息，通过系统的计算，将工资数据从标准输出设备上显示出来。因此我们需要先部署好 C++ 程序的开发环境，编写 C++ 程序，从键盘录入信息并经过计算，最后按要求把员工的工资信息显示出来。

能力目标

通过本项目的学习和开发，学生应该了解什么是 C++ 程序设计；能熟练地部署 C++ 开发环境；能掌握基本的 C++ 语法进行 C++ 编程；能利用运算符和表达式进行运算；能使用基本输入/输出函数实现标准输入/输出设备的输入和输出。

任务 1.1 C++ 入门

任务分析

根据工作情境的要求，要完成实现对各级员工工资的录入、计算和查询等操作，实现工资管理模块，首先必须了解 C++ 程序设计的基本知识、基本语法，掌握 C++ 程序开发环境的配置。

相关知识

1. C++ 程序的结构
2. C++ 程序的书写风格
3. C++ 程序的开发过程
4. C++ 程序的上机环境

实现方法

C++语言是一种高效的、灵活的、既面向过程模型又面向对象模型的优秀高级程序设计语言。通过 C++语言的学习,将有助于我们理解和掌握程序设计的基本理论和技巧,并为后继专业课程的学习夯实基础。

1.1.1 程序和程序语言

“什么是程序?”、“什么是程序语言?”。“程序”一词来自日常生活,指完成某些事务的一种既定方式和过程。从表述方面看,可以将程序看成对一系列动作的执行过程的描述。日常生活中可以看到许多程序方面的例子。如泡茶的过程可以描述为:

- (1) 烧开水;
- (2) 温杯;
- (3) 投茶;
- (4) 洗茶;
- (5) 冲泡。

描述这种程序的就是给出一个包含其中各个基本步骤的序列。如果按顺序实施这些步骤,其整体效果就是完成了该项任务。类似的情况在现实生活中随处可见,当我们身处这些程序性的活动中时,就需要按部就班地逐步完成一系列动作,而对这些动作(事务、活动)过程的细节描述就是一个“**程序**”。

而要完整地描述和说明一个“程序”,则需要对其中每一步细节过程进行说明,包括在程序执行过程中的各种对象及其执行的先后顺序。因此需要有一种恰当的描述和表达方式,我们通常把一套完整的、成体系的表达方式称之为“语言”。

“语言”一词通常指人类在工作与生活中所使用的自然语言,如汉语、英语等,是人们用于交流和表达思想的工具。在前面所描述的现实生活中的程序实例中,我们就是用汉语作为程序的描述语言,所描述的程序是为了给人看、要人去做。

计算机作为人类 20 世纪最伟大的发明,其最重要的功能就是可以根据预先定义的指令,逐步地完成计算工作。为了方便人类控制和指挥计算机工作,需要一种语义清晰、既便于人类理解又能被计算机处理的描述方式,通常将其称为**程序设计语言**,也常被称为编程语言,本书中简称为程序语言或语言。

如同人类有多种自然语言一样,计算机语言也有多种,根据计算机的发展史,程序语言可分为以下几类:

机器语言 是面向机器的,是特定计算机系统所固有的语言。如用机器语言进行程序设计,就需要对机器结构有较深的了解。用机器语言编写的程序可读性很差,也难以维护。

汇编语言 为了提高效率,人们考虑用有助记忆的符号来表示机器指令中的操作码和操作数,例如用 SUB 表示减法、ADD 表示加法、DIV 表示除法等。相对机器语言而言,用汇编语言编写程序的难度有所降低,程序的可读性也有所改观,但仍与人类的自然语言相差甚远。

高级语言 我们知道上述两种语言十分接近,它的书写格式在很大程度上取决于特定计算机的机器指令,这不便于人们的思维和交流。而高级程序语言更接近人们所习惯的描述形式,

而且是与具体机器无关的,这样程序设计者不需要了解机器内部结构,只要按照计算机语言的语法编写程序即可如 Fortran、Pascal、C/C++和 Basic 等。因此用这种语言书写程序的效率会更高,并促使人们去开发更多的应用系统,从而大大推动了计算机应用的发展。可以说,高级程序设计语言的诞生和发展对计算机发展起了极其重要的作用。

1.1.2 C++程序结构

通常我们学习 C++程序设计语言是从一个简单的程序例子来入门的。通过对这个程序的分析使我们对 C++的程序结构有个初步的认识。以下代码是一个完整的 C++程序:

示例 1.1.1 输出一个字符:“欢迎来到 C++的世界!”

程序清单如下:

```
/*  
 * Name: e.1.1.1.cpp  
 * Author:  
 * Description: 教材示例 1.1.1  
 *           在屏幕上显示一行字符:欢迎来到 C++的世界!  
 * Date:2013-01-25  
 * Version: 1.0  
 * Copyright:  
 */  
  
#include <iostream>           //使用标准输入/输出库  
using namespace std;         //使用标准命名空间  
int main(){  
    cout<<"欢迎来到 C++的世界!"<<endl;    //在屏幕上显示"欢迎来到 C++的世界!"  
    system("pause");  
    return 0;                 //结束程序运行  
}
```

以上程序经编译运行后,结果如图 1.1.1 所示。

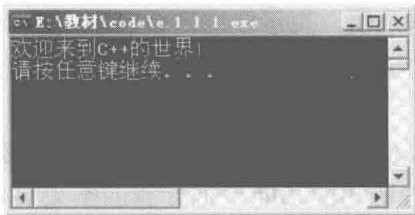


图 1.1.1 运行结果

从上面的程序清单和运行结果来看,显然它是一个简单的 C++程序。但是它具备了一个 C++程序的基本结构,因此我们根据上面的例程来分析一下程序的一般结构。

1. 注释部分

这是关于该程序的功能、作者、书写日期等的说明部分,它不会参与程序的执行,更不会影响程序的运算结果,仅仅是为了提高程序的可读性、可修改性和可维护性而书写的对程序语句的说明性文字,在编译时,编译器将不会编译这些语句。通常 C++程序有两种注释语句:

一种是单行注释,即在注释语句的最前面加上“//”表示单行注释。顾名思义,这种注释的语句只能书写在一行中,不能分行书写。如上面程序中的第9行语句。

另一种是多行注释,即在注释块的最前面加上“/*”符号和在最后加上“*/”表示多行注释。这时编译器就会认为在这两个符号之间的语句都属于注释语句部分。如下所示就是典型的程序描述和注释部分。通常为程序的保证可读性、可维护性和可修改性,在程序首部附加对本程序文件的一些说明文字。该部分的注释主要由如下几个部分组成:

Name: 表示本程序所保存的程序文件名;

Author: 表示书写本程序的作者;

Description: 描述程序所能实现的功能;

Date: 表示书写本程序的日期;

Version: 表示版本号;

Copyright: 申明程序的版权。

```

/*****
 * Name: e.1.1.1.cpp
 * Author:
 * Description: 教材示例 1.1.1
 *           在屏幕上显示一行字符:欢迎来到 C++的世界!
 * Date: 2013-01-25
 * Version: 1.0
 * Copyright:
 *****/

```

2. 预处理命令部分

C++程序的预处理命令以“#”开头,如上面程序中的“#include <iostream>”。目前 C++ 提供了3类预处理命令:宏定义命令、文件包含命令和条件编译命令。上例程序中的“#include <iostream>”语句就属于文件包含命令,表示将标准的 I/O 流的库函数引入到本程序中。

3. 名字空间定义部分

在上例程序中,第10行“using namespace std;”语句的意思是“使用命名空间 std”。C++ 标准库中的类和函数是在命名空间 std 中声明的,因此程序中如果需要用到 C++ 标准库,就需要用“using namespace std;”作声明,表示要用到命名空间 std 中的内容。如果没有使用这个命名空间,系统将提示“cout”和“endl”这两个函数没有申明。

4. 输入/输出

C++程序中总是少不了输入和输出语句,以实现与程序内部的信息交流。特别是屏幕输出功能,使用的频率相当高,这些语句的功能是把计算的结果显示在屏幕上。如上面程序的第12行,即“cout << “欢迎来到 C++的世界!”;”,就是将“欢迎来到 C++的世界!”显示在屏幕上。

5. 函数

通常 C++的程序是由若干个文件组成的,每个文件又是由若干个函数组成,因此,可以说 C++的程序就是函数串,即由若干个函数组成,函数间是相对独立的,并且是并行的,函数间可以相互调用。C++中规定,组成一个程序时,需要有一个并且只能有一个**主函数**,在 C++中将其命名为 main()。执行程序时,系统首先找主函数,并且从主函数开始执行。在标准

C++中规定主函数为整数类型。

6. 语句

语句是组成程序的基本单元。语句由单词组成，单词间用空格符分隔，C++语言中规定语句是以分号结束。如“cout << "欢迎来到 C++的世界!";”。在 C++中有表达式语句、空语句、复合语句、分支语句等若干类。

1.1.3 C++程序的书写格式和风格

C++程序的书写格式基本原则如下：

1. 语句书写

一行一般写一条语句。短语句可以一行写多个；长语句可以一条写多行，但不能将一个单词分开，用双引号引用的一个字符串也最好不分开，如果一定要分开，有的编译系统要求在行尾加续行符（“\”）。

2. 缩进格式

为了提高 C++程序的可读性，一般要求采用适当的缩格书写方式，同一类内容的语句行要对齐，不同层次的内容要有缩进。如在下面程序中，第 4~6 行属于同一类内容，但与第 3、7 行属于不同层次的内容。

```
1 #include <iostream> //使用标准输入/输出库
2 using namespace std; //使用标准命名空间
3 int main(){
4     cout<<"欢迎来到 C++的世界! "<<endl; //在屏幕上显示"欢迎来到 C++的世界! "
5     system("pause");
6     return 0;          //结束程序运行
7 }
```

3. 大括号书写

程序中大括号的书写方法较多。在本书中采用的方法如下：大括号单独一行，并与大括号对应语句对齐，大括号内的语句整体缩进两字符。

4. 注释

为了提高 C++程序的可读性，要求尽量对程序加上注释说明，特别是程序中的类、函数和关键代码部分。

示例代码：

风格不符合规范的代码

```
#include <iostream>
using namespace std;
int main
()
{
    cout<<"欢迎来到 C++的世界! "
    <<endl;
    system("pause");
    return 0;
}
```

风格符合基本规范的代码

```
#include <iostream> //使用标准输入/输出库
using namespace std; //使用标准命名空间
int main(){
    cout<<"欢迎来到 C++的世界!"<<endl;
    //在屏幕上显示"欢迎来到 C++的世界!"
    system("pause");
    return 0;           //结束程序运行
}
```

该例程序很简单，但第一段代码由于书写方法不当，使得阅读起来比较困难，难以分析输出结果；而书写成第二段代码的形式，分析起来就容易多了。

通过上面两个例子对同样一个程序作了两种不同形式的书写，可以看出正确的书写方法会提高程序的可读性，因此，要求读者在书写程序时特别注意这一点。

练习 1.1.1 请根据示例 1.1.1 编写一个 C++ 程序，要求实现在屏幕中输出“C++，我来了！”。

1.1.4 C++程序设计基本流程

C++ 程序从编写代码到形式一个能运行的程序，需要经过源程序编写、程序编译、目标文件链接和生成可执行程序 4 个过程，C++ 程序设计流程如图 1.1.2 所示。

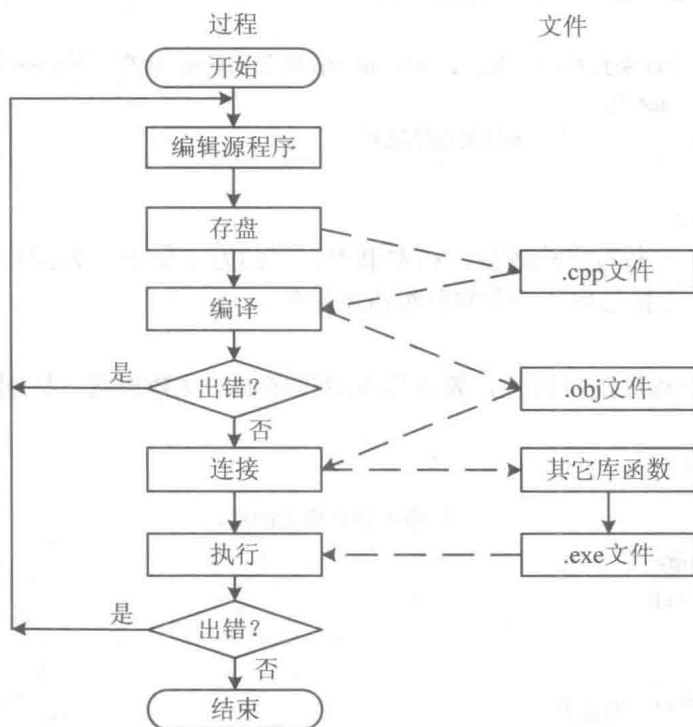


图 1.1.2 C++程序设计流程图

1. 编写 C++ 程序

根据提出的问题，运用编辑工具（如 Dev-C++、VS2008 等）编制解决问题的 C++ 程序。

通常用高级语言编写的程序称为“源程序”(Source Program)。C++中的源程序是以.cpp 作为文件的扩展名的。

2. 编译 C++源程序

在计算机中,机器只能识别和执行由 0 和 1 组成的指令,不能识别和执行用高级语言写的指令。为了使计算机能执行高级语言源程序,必须先用一种称为“编译器(Compiler)”的系统软件,把源程序翻译成机器能识别和执行的二进制形式的“目标程序(Object Program)”。

编译是以源程序文件为单位分别编译的,每一个程序单位组成一个源程序文件,如果有多个程序单位,系统就会分别将他们编译成多个目标程序。目标程序一般以.obj 或.o 作为扩展名。编译的作用是对源程序进行语法和词法检查。语法检查是根据源程序的上下文来检查程序的语法是否有错,例如在 cout 语句中输出事先没有定义的变量。编译时对文件中的全部内容进行检查,编译结束后会显示出所有的编译出错信息。而词法检查是检查源程序中的单词拼写是否出错,例如把 main 拼成 mani。通常编译系统在编译时会给出两类出错信息:一种是错误信息(error),这是一种致命错误,将导致程序无法完成编译,也不会生成目标程序;另一种是警告信息(warning),这种出错是一些不影响运行的轻微的错误,编译系统可以为源程序生成目标程序。

3. 将目标文件链接

在改正所有的错误并通过编译后,将得到一个或多个目标文件。此时的目标文件还不能直接在机器上运行,还需要用系统提供的“连接程序(Linker)”将一个程序的所有目标程序和程序中所用到的库文件以及系统提供的其他信息连接起来,最终生成一个可执行的二进制文件,它的扩展名为.exe。

4. 运行程序

运行生成的可执行的二进制文件,得到运行结果。

5. 分析运行结果

如果运行结果与所预期的不一致,则应检查程序或算法是否有错。

上述过程如图 1.1.2 所示。其中实线表示操作流程,虚线表示文件的输入和输出。例如,编辑源程序后得到源程序文件 e.1.1.1.cpp,然后在进行编译,时将源程序文件 e.1.1.1.cpp 输入,经编译,如没有错误出现,则生成目标文件 e.1.1.1.obj,如出错,则返回到第 1 步进行程序编辑以改正错误;当生成目标文件 e.1.1.1.obj 后,再将该文件与系统提供的库文件等连接,生成可执行文件 e.1.1.1.exe;最后把 e.1.1.1.exe 调入内存并运行,分析运行后得到的结果,如有错,返回到第 1 步重新检查源程序算法,如正确,则该程序开发过程结束。

1.1.5 C++开发环境简介

C++是一门实践性要求很高的课程,在了解 C++语言的初步知识后,接下来我们就要在计算机上亲自动手去编写和运行 C++程序,以达到深入理解 C++程序和初步掌握 C++上机操作的目的。

正如我们前面提到的 C++程序开发过程中包含编辑和编译两个过程,因此在上机实践中,我们实际上要使用的软件也分为两类:一类是编辑工具,用于程序的编写;另一类是编译工具,用于将源程序转变成可执行程序。而为了方便程序员的开发,往往开发工具提供商将编辑工具

和编译工具集成起来，称之为 IDE（集成开发环境）。

目前市场上经常使用 C++ 开发环境的有 Microsoft 公司的 Visual C++（简称 VC），Borland 公司的 C++ Builder（简称 BCB），但是这两种开发工具对以学习 C++ 语言为目的的初学者来说较为复杂，因此在本书的学习中，我们将以共享软件 Dev-C++ 作为主要的开发工具来介绍 C++ 语言，有兴趣的学生可以自己在课后去使用 VC 或 BCB。下面我们将从 Dev-C++ 的安装开始介绍 Dev-C++ 的使用。

Dev-C++ 简介

Dev-C++ 是一个遵守 GNU 条款的开源软件（Open Source Software），包括一个多页面窗口的工程编辑器，其中集合了编辑器、编译器、链接程序和执行程序，是一个全功能的 C++ 开发环境。而且由于其使用 GCC 作为编译器，对 C++ 的标准支持比 VC 要好，比较适合初学者学习使用。

Dev-C++ 软件下载

本书示例代码运行环境为 Dev-C++ 4.9.9.2 版，该软件可以在网络上免费下载，如天空软件。

Dev-C++ 安装

Dev-C++ 的安装程序与其他软件的安装完全一致，可根据下面图例逐步完成。

（1）双击安装文件，首先出现安装语言选择（如图 1.1.3 所示），默认并单击“OK”按钮，然后出现版权许可画面（如图 1.1.4 所示），单击“I Agree”按钮。

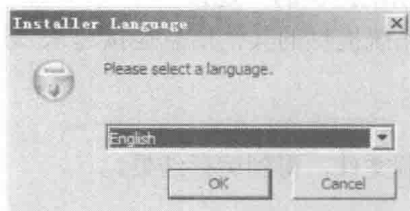


图 1.1.3 Dev-C++ 安装—语言选择

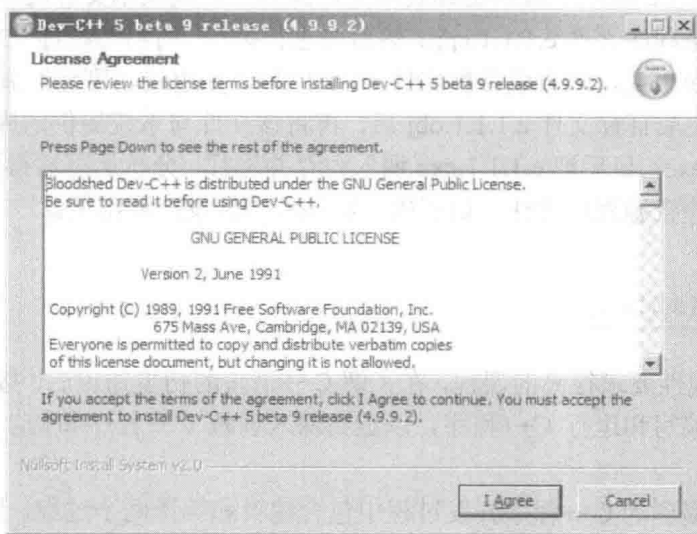


图 1.1.4 DevC++ 安装—版权信息

(2) 出现组件选择界面 (如图 1.1.5 所示), 这里可以选择不同的安装方式。

Full: 全部安装, 安装所有程序组件、范例、图片、帮助和不同语言界面。

Typical: 典型安装, 除不同语言界面外, 其余全部安装。

Custom: 用户自定义安装, 在典型安装的基础上, 可自行决定要安装的文件数量。

确定要安装的类型后, 单击 “Next >” 按钮。

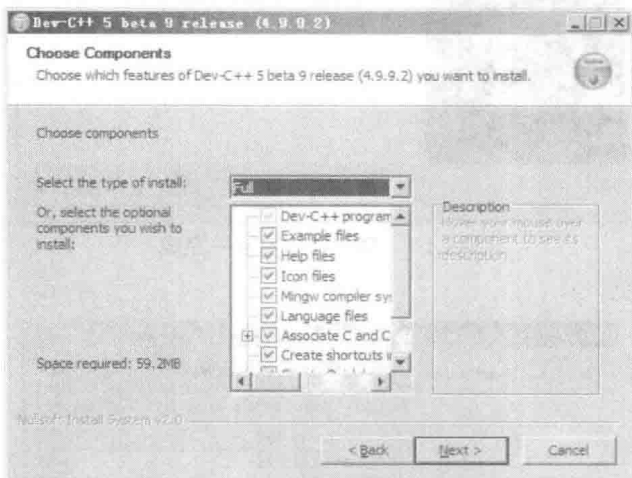


图 1.1.5 Dev-C++安装—系统配置

(3) 选择安装目录, 在文本框中输入文件的安装目录, 也可通过单击 “Browse” 按钮进行选择 (如图 1.1.6 所示)。确定目录后, 单击 “Install” 按钮进行安装。

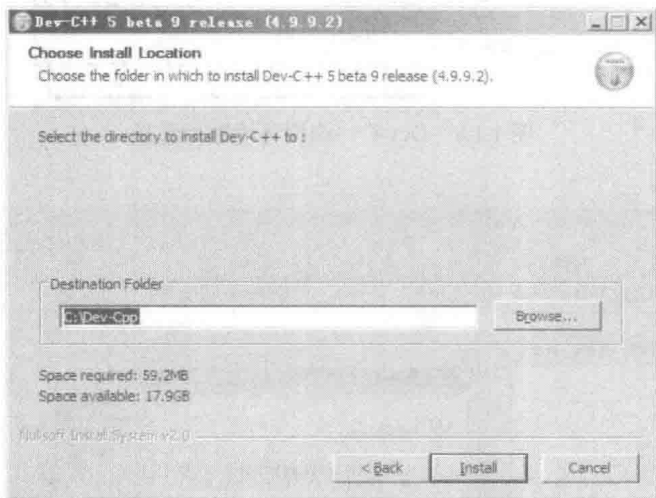


图 1.1.6 Dev-C++安装—存储地址设置

(4) 安装完成后, 出现安装结束界面, 如图 1.1.7 所示。

使用 Dev-C++

(1) 启动 Dev-C++, 第一次启动界面。

启动 Dev-C++后, 一般会出现选择语言选择界面, 如图 1.1.8 所示, 选择 Chinese, 单击 “Next” 按钮后进入第一次启动界面, 如图 1.1.9 所示; 可以看到界面变成中文界面, 同时出现每日提示。



图 1.1.7 Dev-C++安装—完成

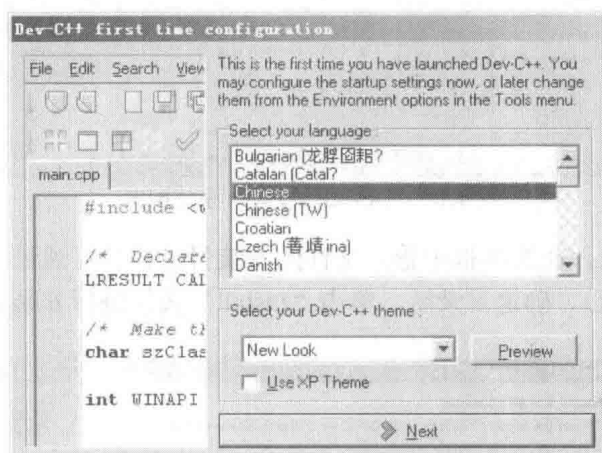


图 1.1.8 Dev-C++编辑环境语种选择

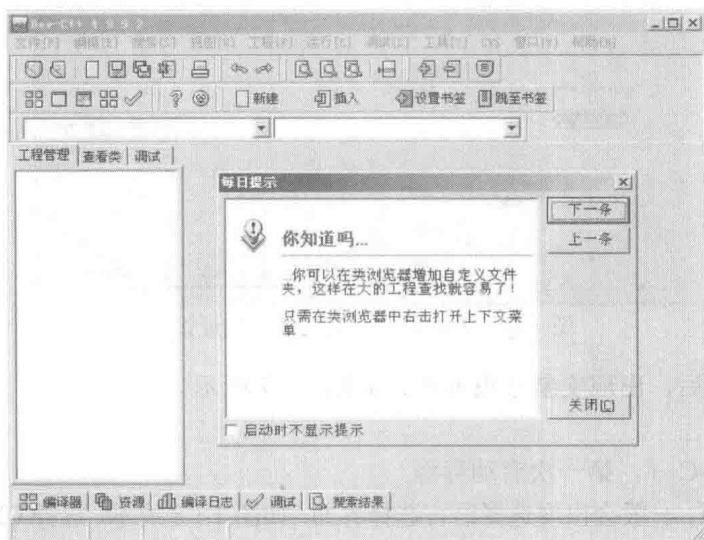


图 1.1.9 Dev-C++启动界面

- 关闭每日提示

大家可以先将“Don't Display tips at startup”（启动时不显示提示）选中，下次启动就不会出现这样的帮助提示。

- 菜单中转换中文界面

Dev-C++在4.8以后的版本中提供了中文界面（如图1.1.10所示），具体设置如下：

请选择 Tools（工具）→Environment Options（环境选项）→Interface（界面）命令，在 Language（语言）下拉列表框中将 English（Original）改为 Chinese，然后单击“OK”按钮即可。

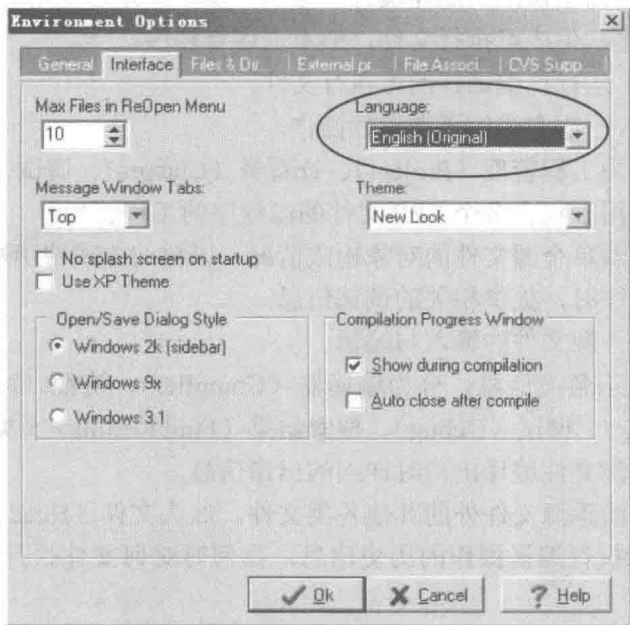


图 1.1.10 Dev-C++界面语种选择

（2）Dev-C++的操作界面简介。

下面我们对整个窗口作一个简单的介绍，打开 e.1.1.1.cpp 程序后，就可以进行如图 1.1.11 所示的 Dev-C++编辑界面。

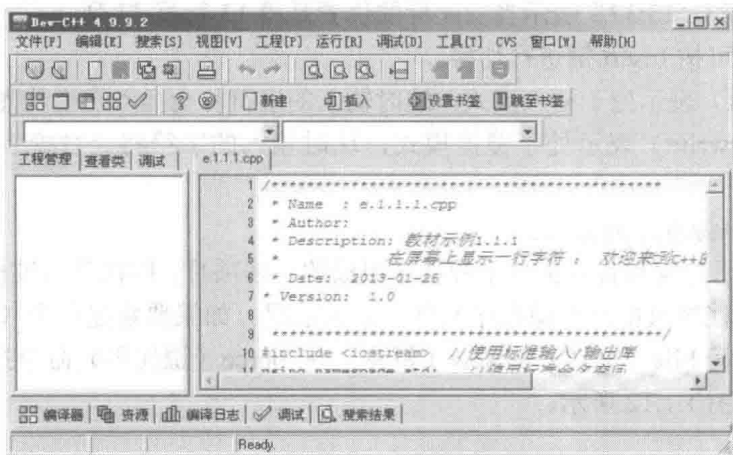


图 1.1.11 Dev-C++编辑界面