第1章 初识C++

为了解决软件危机，20 世纪 80 年代计算机界提出了面向对象程序设计（Object Oriented Programming，OOP）的编程思想，支持面向对象程序设计的语言也应运而生。其中，C++ 语言是最流行的一种面向对象程序设计语言。

C++ 语言在计算机编程语言中占有极其重要的地位，可以说是所有计算机程序设计语言中非常伟大的发明之一。C++ 语言自 1983 年问世以来，一直备受程序员的青睐，是非常受欢迎的编程语言之一。C++ 语言在 TIOBE 2019 年 5 月公布的编程语言排行榜[①]中排名第三位，而且在近 20 年（2001—2019 年）的时间里始终稳定地位居前三位，如图 1-1 所示，足见 C++ 语言的魅力。

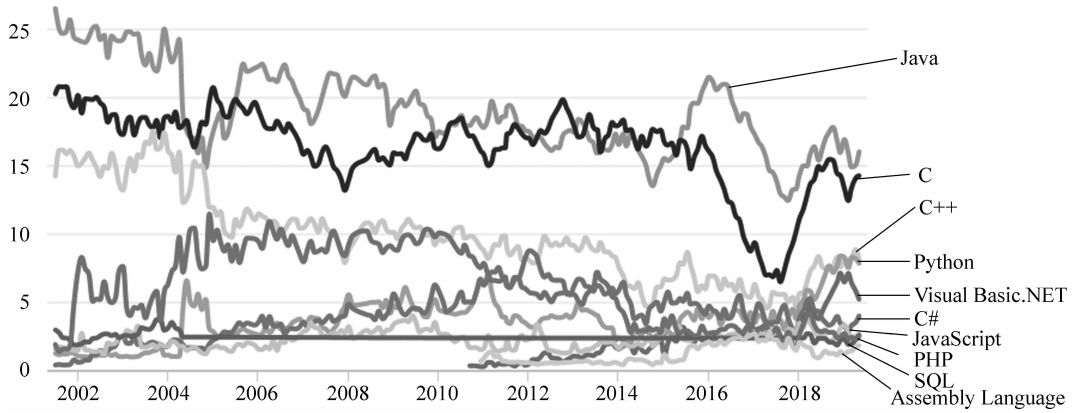

图 1-1 TIOBE 编程语言排行榜

接下来，让我们一起揭开 C++ 语言的神秘面纱，领略 C++ 语言的独特魅力吧！

1.1 C++ 简介

1.1.1 C++ 发展史

计算机科学中对象和实例的概念最早萌芽于麻省理工学院的 PDP-1 系统，而 Simula67 被认为是最早的面向对象程序设计语言，它引入了所有后来面向对象程序设计语言所遵循

① TIOBE 编程语言排行榜根据互联网上有经验的程序员、课程和第三方厂商的数量，并使用搜索引擎（如 Google、Bing、Yahoo!）Wikipedia、Amazon、YouTube 统计出排名数据，只是反映某个编程语言的热门程度，并不能说明一门编程语言好不好，或者一门语言所编写的代码数量多少，其结果作为当前业内程序开发语言的流行使用程度的有效指标。

的基础概念：对象、类、继承等。随后出现了 Smalltalk 语言，被公认为历史上第二个面向对象程序设计语言和第一个真正的集成开发环境（Integrated Development Environment，IDE）。Smalltalk 对其他众多的程序设计语言（如 Objective-C、Java 和 Ruby 等）的产生起到了极大的推动作用。1982 年，美国 AT&T 公司贝尔实验室的本贾尼·斯特劳斯特卢普（Bjarne Stroustrup）博士在 C 语言的基础上引入并扩充了面向对象的概念，发明了一种新的程序设计语言。为了表达该语言与 C 语言的渊源关系，1983 年该语言被正式命名为 C++（C Plus Plus），而本贾尼·斯特劳斯特卢普博士被尊称为“C++ 语言之父”。

自从 C++ 语言被发明以来，它经历了多次修订，每一次修订都为 C++ 语言增加了新的特征并做了一些修改。第一次修订是在 1985 年，在此期间本贾尼·斯特劳斯特卢普博士出版了他的经典巨著 *The C++ Programming Language*，这时 C++ 语言已经开始受到关注。1990 年，本贾尼·斯特劳斯特卢普博士又出版了一部传世经典 *The Annotated C++ Reference Manual*（简称 ARM），由于当时还没有 C++ 语言标准，ARM 成为事实上的标准。1990 年，在 C++ 语言中引用了模板（Template）和异常（Exception），使 C++ 语言具备了泛型编程（Generic Programming）和更好的运行期错误处理方式。1991 年，负责 C++ 语言国际标准化的技术委员会工作组召开了第一次会议，开始进行 C++ 语言国际标准化的工作。从此，美国国家标准学会（American National Standards Institute，ANSI）和国际标准化组织（International Standards Organization，ISO）的标准化工作保持同步，互相协调。1993 年，运行期类型识别（Run-Time Type Identification，RTTI）和名称空间（Namespace）加入 C++ 语言中。1994 年，C++ 语言的第一个标准化草案出台。在完成 C++ 语言标准化的第一个草案后不久，亚历山大·斯特帕诺夫创建了标准模板库（Standard Template Library，STL），STL 不但功能强大，而且非常优雅，标准化委员会投票并通过了将 STL 包含到 C++ 语言标准中的决议，于 1997 年通过了该标准的最终草案。1998 年，C++ 语言的 ANSI/ISO 标准被投入使用。通常，这个版本的 C++ 语言被认为是标准 C++ 语言，称为 C++ 98。2003 年，ISO 的 C++ 语言标准化委员会又对 C++ 语言略做了一些修订，发布了 C++ 03 语言标准，这个新版本是一次技术性修订，对第一版进行了整理，如修订错误、减少多义性等。2005 年，一份名为 *Library Technical Report 1*（TR1）的技术报告发布，为 C++ 语言加入了正则表达式（Regular Expression）、哈希表（Hash Table）等重要类模板。ISO 于 2011 年发布 C++ 11 标准，取代 C++ 98 和 C++ 03。C++ 11 对 C++ 语言的语言特性和标准库都做了比较大的扩充，TR1 中的许多特性正式成为 C++ 11 语言标准的一部分。2014 年，ISO 发布了 C++ 14 语言标准，C++ 14 旨在作为 C++ 11 的一个小扩展，主要提供漏洞修复和小的改进。C++ 14 是 C++ 11 的增量更新，主要是支持普通函数的返回类型推演（Return Type Deduction）、泛型 lambda、对 constexpr 函数限制的修订、constexpr 变量模板化等。2017 年，ISO 发布了 C++ 17 语言新标准，引入了许多新的语言特性，如用于可变参数模板的折叠表达式（Fold Expressions）、内联变量（Inline Variables）、类模板参数规约（Class Template Argument Deduction）等。目前，最新标准 C++ 20 制订工作正在推进中。

本书中介绍的语法遵循 C++ 11 语言标准，主要基于以下两个原因：一是 C++ 11 相较于 C++ 98 做了较大的改进，引入了很多新特性，这使得 C++ 语言感觉像是一门“新”的语言；二是本书的定位是面向对象程序设计的入门级教材，C++ 11 以后的标准虽然也增加了很多新特性，但是这些新特性对于面向对象程序设计的初学者来说过于深奥。还有一点原因是，任何一款 IDE 都需要对新标准有一个适应改进的过程，有些 IDE 对最新标准未必全

部提供支持。本书重点介绍大多数编译器都支持的特性。

1.1.2　C++ 应用领域

C++ 语言兼容 C 语言，是一种接近计算机硬件编程的语言，因此在系统级的开发上，C/C++ 语言应用居多。C++ 语言应用领域主要集中在以下几个方面。

(1) 嵌入式系统开发。C++ 语言是一种功能强大的面向对象的程序设计语言，在嵌入式系统开发中使用 C++ 语言，会获得意想不到的简洁和喜悦。使用 C++ 语言进行程序设计具有很高的效率，而且 C++ 语言对 C 语言的兼容性使得底层平台的设计也很高效，同时具有很大的灵活性，因此使得它在底层开发中有着极大的应用。

(2) 游戏开发。近年来，C++ 语言凭借先进的数值计算库、泛型编程等优势，在游戏设计领域有较多的应用，绝大部分的游戏引擎是使用 C++ 语言编写的，如 UE4。除了一些网页游戏外，很多游戏客户端程序都是基于 C++ 语言开发的。

(3) 虚拟现实和仿真。虚拟现实(Virtual Reality，VR)是一种可以创建和体验虚拟世界的计算机仿真系统，是利用计算机生成的一种实时动态的三维立体逼真图像，结合 VR 眼镜，可以在观影、游戏、旅游活动、教学等方面给人一种完美的沉浸式体验。C++ 语言在这一技术中扮演着重要的角色，常规的 VR 引擎基本上用 C++ 语言开发。

(4) 网络软件开发。C++ 语言拥有大量成熟的用于网络通信的库，自适应通信环境(Adaptive Communication Environment，ACE)是其中最具有代表性的跨平台库，在许多重要的企业部门甚至是军方都有应用。知名通信软件 QQ 核心代码就是基于 C++ 语言编写的。

(5) 数字图像处理。在数字图像处理领域中，基于 C++ 语言开发的程序占有很大的比例，如 OpenCV 视觉识别技术。

当然，C++ 语言的应用远不止这些。随着信息化、智能化、网络化的发展，大数据计算、人工智能的不断发展及应用，C++ 语言的应用会越来越广泛，在各个应用领域都将发挥重要的作用。

1.2　C++ 程序集成开发环境

C++ 程序设计需要使用合适的集成开发环境。集成开发环境是用于提供程序开发环境的应用程序，一般包括代码编辑器、编译器、语法检查器、调试器和图形用户界面工具等，集成了代码编写功能、编译功能、语法检查功能、调试功能等一体化的开发软件服务套。所有具备这一特性的软件或者软件套(组)都可以称为集成开发环境，如微软公司的 Visual Studio 系列，Borland 公司的 C++ Builder、Code::Blocks 等。Visual Studio 是目前比较流行的 Windows 平台应用程序的集成开发环境，是由美国微软公司设计开发的一套完整的开发工具集，它所写的目标代码适用于微软支持的所有平台，包括 Microsoft Windows、Windows CE、.NET Framework、.NET Compact Framework 等。Visual Studio 功能强大，支持多种编程语言，安装使用 Visual Studio 需要耗费大量计算机资源，不适用于 C++ 语言的初学者。

C++ Builder 是由 Borland 公司推出的一款可视化集成开发工具。C++ Builder 具有快

速的可视化开发环境，可以快速地建立应用程序界面，它实现了可视化的编程环境和功能强大的 C++ 语言的完美结合。但是，本书旨在向读者介绍 C++ 语言基础语法，通过构建控制台应用程序讲解语法知识。从这个角度出发，Code::Blocks 是最佳选择。本书所有代码都在 Code::Blocks 开发环境下编写并通过测试。

1.2.1 Code::Blocks 简介

Code::Blocks 是一个开放源码的全功能的跨平台 C/C++ 语言集成开发环境，最初的开发重点是 Windows 平台，后期又提供对 GNU/Linux 的支持。Code::Blocks 在 1.0 发布时就成为跨越平台的 C/C++ 语言集成开发环境。Code::Blocks 提供了许多工程模板，包括控制台应用、DirectX 应用、动态链接库、FLTK 应用、GLFW 应用、Irrlicht 工程、OGRE 应用、OpenGL 应用、QT 应用、SDCC 应用、SDL 应用、SmartWin 应用、静态库、Win32 GUI 应用、wxWidgets 应用、wxSmith 工程，另外它还支持用户自定义工程模板。在 wxWidgets 应用中选择 UNICODE 支持中文。Code::Blocks 支持语法彩色醒目显示，支持代码补全，支持工程管理、项目构建、程序调试等。

1.2.2 Code::Blocks 环境设置

通常不需要对 Code::Blocks 开发环境做任何配置，直接使用它的默认设置即可完成 C++ 程序的设计。如果需要，可以修改 Code::Blocks 的开发环境。

1. 设置 C++ 语言标准

选择“设置(Settings)”→“编译器(Compiler)”命令，在弹出的对话框中选择相应的语言标准。本书遵循 C++ 11 标准，所以选择 ISO C++ 11，如图 1-2 所示。

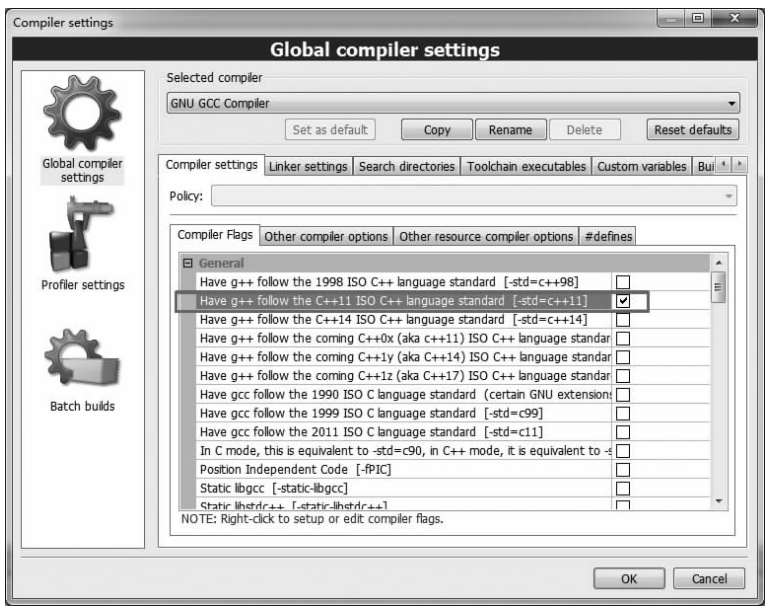

图 1-2 选择语言标准

2. 设置调试器

使用调试器可以调试程序，用于发现程序设计中的错误。Code::Blocks 的调试器需要与编译器匹配，如 MinGW 与 GDB 匹配。首先确定编译器的类型，本书使用 MinGW 编译器，如图 1-3 所示。然后选择“设置(Settings)”→“调试器(Debugger)”命令，在弹出的对话框中设置调试器，如图 1-4 所示。

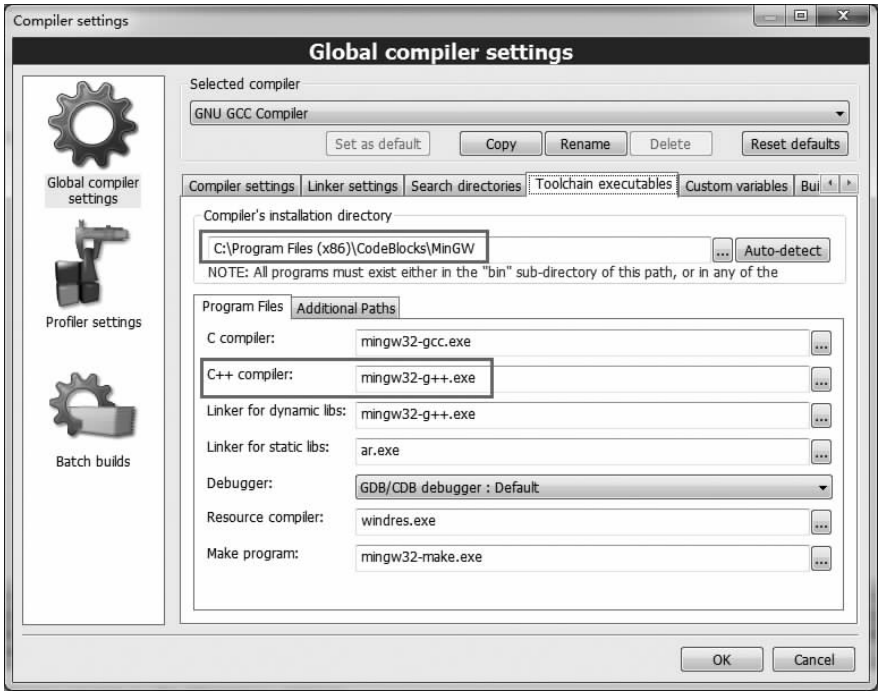

图 1-3　选择编译器

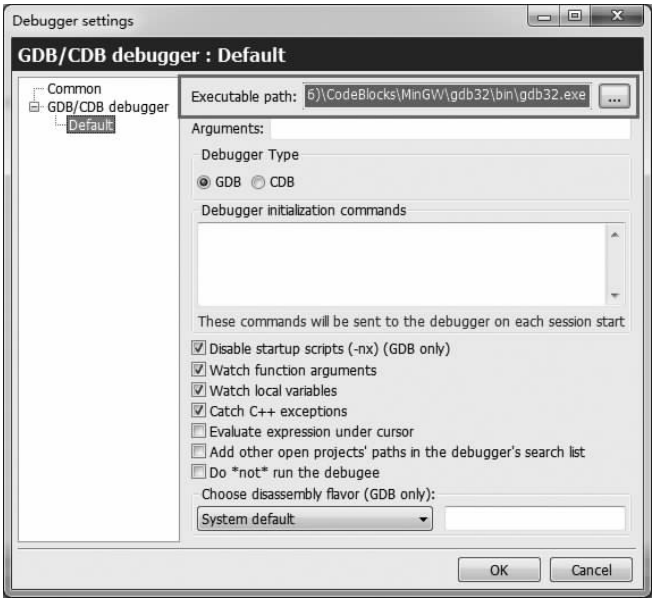

图 1-4　设置调试器

本书中，Code::Blocks 的调试器安装在 C:\Program Files（x86）\CodeBlocks\MinGW\gdb32\bin\gdb32.exe 中。

注意

只有项目文件才能进行程序调试，单源文件不能进行调试。

1.3 C++ 程序设计基本过程

C++ 程序设计需要经过四个步骤：编辑源代码、编译、链接和执行，但具体的步骤取决于计算机环境和使用的 C++ 编译器。在 Code::Blocks 环境中编写 C++ 程序的基本步骤如下。

1. 编辑源代码

使用文本编辑器编写程序，并将其保存为文件，该文件就是程序的源代码文件，简称源文件(Source File)。可以选择任意文本编辑器编写源文件，如记事本。当然，最明智的选择是使用 IDE 自带的编辑器编写源文件。使用 Code::Blocks 编写程序时可以选择创建项目文件或者单源文件。

(1) 创建项目文件

选择“文件(File)”→“新建(New)”→“项目(Project)”→“控制台应用程序(Console application)”命令(注：本书所有示例均创建控制台应用程序)。根据提示完成项目创建，在 main.cpp 文件中编辑源代码。

项目包含很多有用的信息，而且调试程序需要在项目中进行。但是，创建项目文件的操作过程相对复杂，如果仅仅希望通过程序练习 C++ 语言的基本语法，可以创建一个普通的源文件。

(2) 创建单源文件

选择“文件(File)”→“新建(New)”→“空文件(Empty File)”命令，打开文件编辑窗口，默认文件名是 untitled。源文件需要保存才能继续编译，C++ 源文件的扩展名通常是.cpp。当然，不同的操作系统及集成开发环境下创建的 C++ 源文件扩展名有所不同，使用什么扩展名取决于 C++ 语言实现。表 1-1 所示为源文件常用扩展名。

表 1-1 源文件常用扩展名

C++ 实现	源文件扩展名
UNIX	.C、.cc、.cxx、.c
GNU C++	.C、.cc、.cxx、.cpp、.C++
Borland C++	.cpp
Microsoft Visual C++	.cc、.cxx、.cpp
Code::Blocks	.cc、.cxx、.cpp

2. 编译

创建好项目后，需要对程序进行编译。早期的编译器使用一个从 C++ 语言到 C 语言的编译器程序，没有开发直接的 C++ 语言到目标代码的编译器，它把 C++ 源代码翻译成 C 源

代码，然后使用一个标准 C 语言编译器对其进行编译。随着 C++ 语言的普及，越来越多的实现转向创建 C++ 语言编译器，直接把 C++ 源代码生成目标代码。这种方法加速了编译过程，并强调 C++ 是一种独立的语言。

通常，IDE 都提供了编译命名，如 Code::Blocks 中的“建立(Build)”→“编译当前文件(Compile Current File)”命令。还有一些 IDE 提供了诸如“建立(Build)”“生成(Make)”等命令，以完成程序的编译。

如果程序中存在语法错误，语法检查器能够检查出这些错误并给予提示。必须修改程序的所有错误，源文件才能编译成功。如果程序没有语法错误，编译器将生成一个扩展名为.o 的目标代码(Object Code)文件。

3. 链接

链接是指把目标代码同使用的函数的目标代码及一些标准的启动代码(Startup Code)组合起来，生成可执行程序的过程。C++ 程序通常使用库函数，如计算平方根的函数 sqrt()，链接的任务就是把 sqrt()函数的目标代码与当前程序组合为一个整体。通常，IDE 提供了“建立(Build)”命令可完成程序的链接，如在 Code::Blocks 中可以使用“建立(Build)”命令生成可执行程序。

4. 执行

程序执行是指运行生成的可执行程序并得到程序结果的过程。如果程序的输出结果不是预期的正确结果，需要重新修改程序并重新编译、链接生成新的可执行程序，必要时需要调试程序找到错误原因并修改程序。在 Code::Blocks 中可以使用“执行(Run)”命令生成可执行程序。

C++ 程序设计的大致过程如图 1-5 所示。

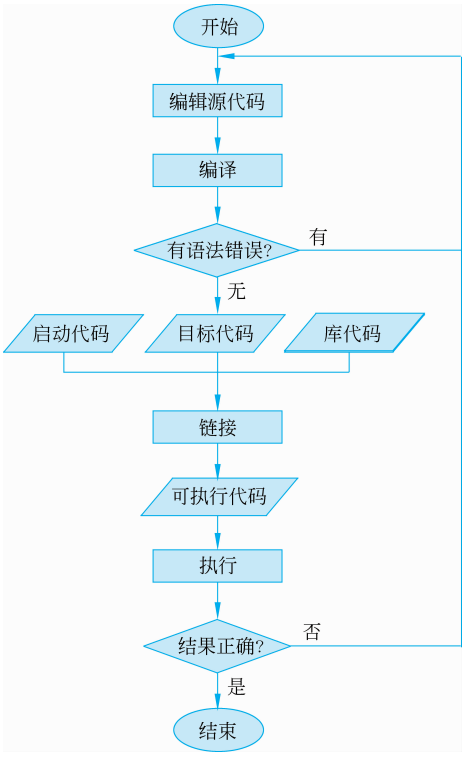

图 1-5　C++ 程序设计步骤

1.4 第一个程序

创建第一个程序：设计控制台应用程序，在显示器上输出“Hello world.”（注：双引号不是输出内容的一部分）。有意思的是，几乎所有语言的第一个示例程序都是向显示器输出“Hello world.”。

【例 1-1】

```
Line 1   #include <iostream>                    //文件包含预处理命名
Line 2   using namespace std;
Line 3   /*
Line 4   函数功能:在显示器上输出"Hello world."
Line 5   参数:无
Line 6   返回值:整数 0
Line 7    */
Line 8   int main()
Line 9   {
Line 10      cout <<"Hello world." <<endl;
Line 11      return 0;
Line 12  }
```

程序运行结果如图 1-6 所示。

图 1-6 例 1-1 程序运行结果

1.4.1 C++ 程序基本结构

C++ 程序是由一个或多个文件构成的，这些文件可能是源文件或头文件等。源文件是由一个或多个函数构成的，一个程序有且仅有一个名字为 main()的称为“主函数”的函数。函数的基本结构如下：

```
类型说明符 函数名(参数列表)
{
    语句序列;
}
```

其中，第一行称为函数头或函数首部，大括号部分称为函数体，分号是语句结尾的标志。

【例 1-2】

```
Line 1   #include <iostream>
Line 2   #include <cmath>
Line 3   using namespace std;
Line 4   bool isPrime(int n)
Line 5   {
Line 6       int k =sqrt(n);                    //整型变量 k 等于变量 n 的平方根
Line 7       for(int i =2; i <=k; i++)
Line 8       {
Line 9           if(n %i ==0) return false;
Line 10      }
Line 11      return true;
Line 12  }
Line 13  int main()
Line 14  {
Line 15      int intNum;
Line 16      cout <<"请输入一个整数:";
Line 17      cin >>intNum;                      //从键盘输入一个整数
Line 18      if(isPrime(intNum) ==true)
Line 19      {
Line 20          cout <<intNum <<"是素数。" <<endl;
Line 21      }
Line 22      else
Line 23      {
Line 24          cout <<intNum <<"不是素数。" <<endl;
Line 25      }
Line 26      return 0;
Line 27  }
```

程序的功能：从键盘输入一个整数，判断该整数是否为素数，并输出判断结果。程序运行结果如图 1-7 所示。

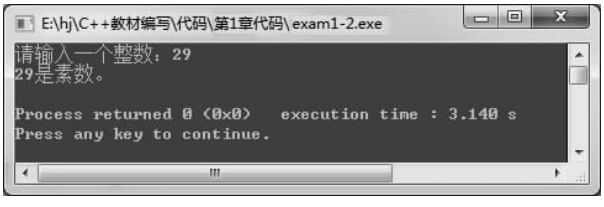

图 1-7　例 1-2 程序运行结果

本例包含两个函数：main()函数和 isPrime()函数，在 main()函数中调用了 isPrime()

函数(Line 18)。C++语言的函数可以给调用者返回一个值,该值称为返回值(Return Value),值的类型就是返回值的类型。例如,isPrime()函数返回一个bool类型值(Line 4),main()函数返回一个整型值(Line 13)。

1. 关于main()函数

来看一下main()函数的接口描述。int是主函数main()的返回值类型,即main()函数返回一个整数值。接下来是空括号,意味着main()函数不接收任何信息,或者说main()函数不接收任何参数(Argument)。当然,并不是说main()函数必须不带任何参数,事实上main()函数的括号内可以带有形参,例如:

```
int main(int argc, char * * argv)
{
    cout <<"Hello world." <<endl;
    return 0;
}
```

相信已经学习过C语言的读者对main()函数的形参意义及作用非常了解,本书不再讨论这些参数的意义。

main()函数的最后通过语句"return 0;"返回整数0,这是非常有意义的工作,因为程序员可以通过程序的返回值来判断程序运行是否正常结束。如果程序正常结束,main()函数返回0(如图1-7所示,"Process returned 0..."):如果程序因为某种原因非正常结束,main()函数将返回一个非零值,例如:

```
#include <iostream>
using namespace std;
int main()
{
    int * p =new int[-1];
    return 0;
}
```

以上代码的运行结果如图1-8所示。

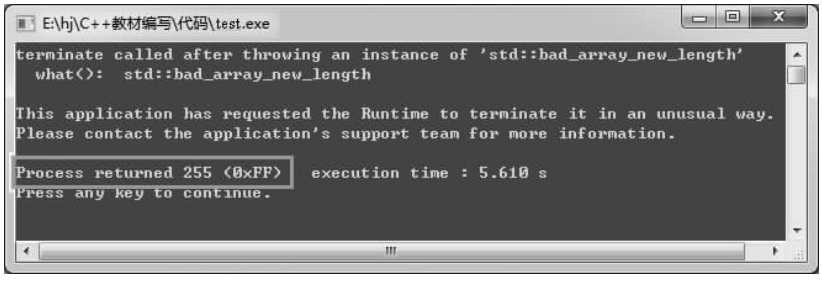

图1-8 程序非正常结束

程序返回了一个非零随机整数(255),说明程序并没有正常结束。

熟悉C语言编程风格的读者可能会使用如下main()函数。