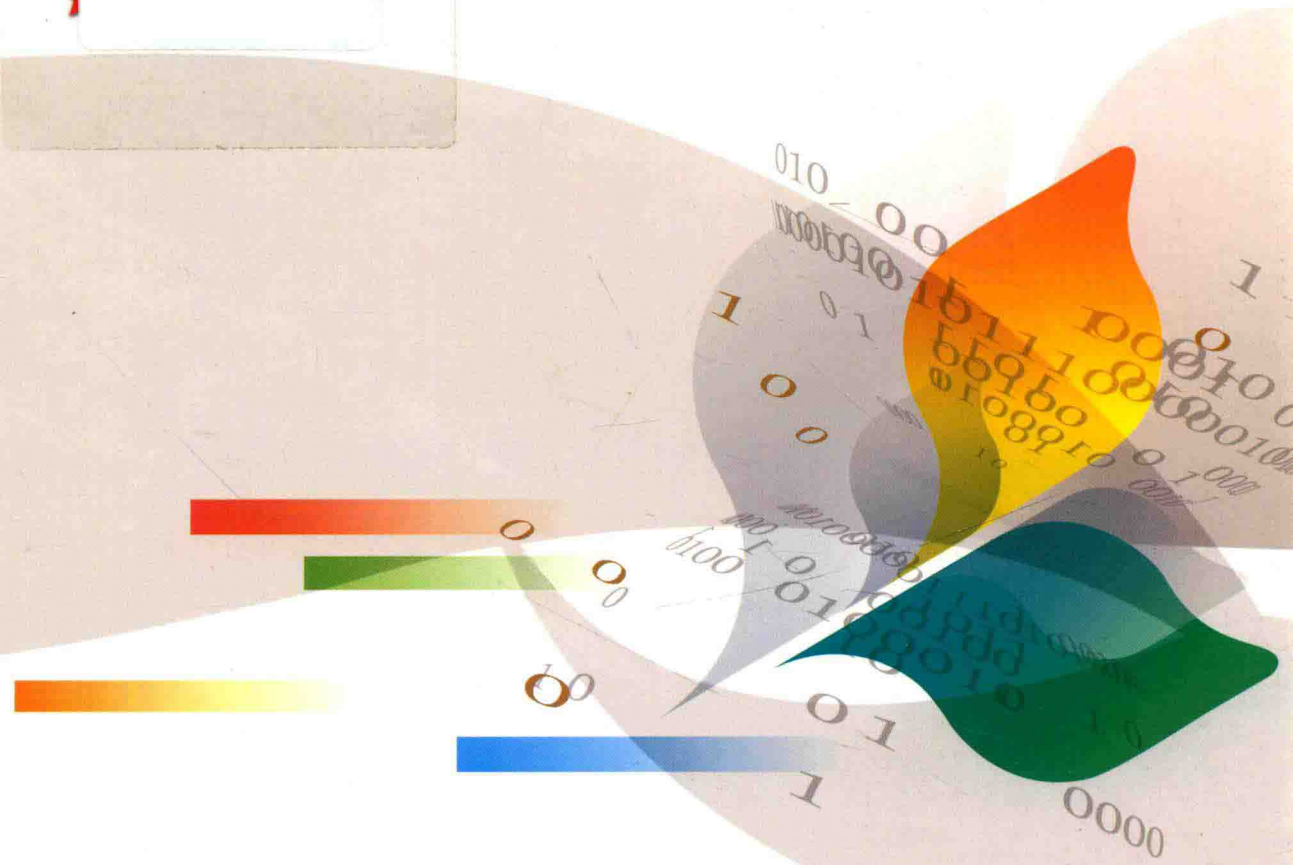
教育“十三五”规划教材



C++程序设计

（基于C++11标准）

◎ 李长河　童恒建　叶亚琴　杨 鸣　编著

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

普通高等教育“十三五”规划教材
中央高校教育教学改革基金(本科教学工程)资助

C++程序设计

(基于C++11标准)

李长河　童恒建　叶亚琴　杨　鸣　编著

電子工業出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

本书紧随C++发展的步伐，立足于培养工程实践能力强、创新能力强、具备国际竞争力的高素质复合型“新工科”人才，是全面采用C++11新标准编写的面向对象程序设计的教材。本书不仅讲解新标准下C++的基本语法，展现C++的发展现状，更注重编程思维和解决实际问题的能力的培养。本书结合算法与数据结构，通过简明的例子讲解C++的特性和使用方法。内容涵盖新标准下C++基本语法、面向对象程序设计核心技术、基本数据结构，以及常用算法和工具。

本书适用于高等学校理工科各专业的C++程序设计、面向对象程序设计等课程，也可供初级或中高级程序员及工程技术人员参考使用。

图书在版编目(CIP)数据

C++程序设计：基于C++11标准/李长河等编著. —北京：电子工业出版社，2018.8
ISBN 978-7-121-34352-0

Ⅰ. ①C… Ⅱ. ①李… Ⅲ. ①C++语言—程序设计—高等学校—教材 Ⅳ. ①TP312.8

中国版本图书馆CIP数据核字(2018)第115542号

策划编辑：戴晨辰
责任编辑：张 京　　特约编辑：张燕虹
印　　刷：北京京师印务有限公司
装　　订：北京京师印务有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路173信箱　　邮编：100036
开　　本：787×1092　1/16　印张：17.5　　字数：448千字
版　　次：2018年8月第1版
印　　次：2018年8月第1次印刷
定　　价：55.00元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010)88254888，88258888。

质量投诉请发邮件至zlts@phei.com.cn，盗版侵权举报请发邮件至dbqq@phei.com.cn。

本书咨询联系方式：dcc@ phei. com. cn。

Preface

为什么要写这本书

C++作为一种通用程序设计语言，支持数据抽象、面向对象编程、泛型编程及底层的内存管理，且兼容C语言，是系统编程、桌面应用、服务器软件、嵌入式系统、游戏、实时系统、高性能计算等领域首选的编程语言，也是人工智能及其各分支领域最受欢迎的编程语言之一。C++是高等学校理工科专业普遍开设的具有很强工程实践性的一门课程，也是程序设计课程的主流。

近年来，C++有了很大的发展，尤其是C++11新标准的发布，给C++带来了革命性变化，正如C++创始人Bjarne Stroustrup所说“C++11看起来像一门新的语言”。C++11变得更加现代化和智能化，大幅度扩充了C++ 98标准下的语言功能，如类型自动推导、lambda表达式、并行编程、移动语义等，这些变化体现了C++与时俱进的活力，它们不仅显著提高了程序的开发效率和性能，也使得语言更为统一、易于教学。

自从C++11新标准发布以来，各大编译器厂商和组织纷纷跟进对C++11的支持。在编写本书的时候，大部分主流的C++编译器已经全面支持C++11，比如GCC、Microsoft Visual C++、Intel C++、Clang等编译器。这给读者学习C++11提供了良好的编程环境。因此，无论从C++本身发展的角度还是从业界对C++需求的角度，学习和使用C++11是大势所趋。

读者对象

本书包含C语言基本知识，因此本书不仅适合已经有C语言基础的读者，也适合没有C语言基础的读者。编者认为，没有必要为了学习C++而先去学习C语言。虽然C++兼容C语言，但C++中以面向对象为基础的编程风格和C语言中以面向过程为基础的编程风格是有明显区别的。C语言的编程习惯会影响C++的学习，编者认为C++可以作为入门语言来学习。

另外，如果读者已经学习了 C++98 标准下的语法，通过本书可以快速掌握 C++11 新标准下的核心语言特性，对原有 C++ 程序进行结构改造和性能升级。

本书特色

在培养“新工科”人才的时代要求和中国智能制造发展战略背景下，本书旨在培养读者的编程思维和提高读者的工程实践能力，通过简明的例子，结合常用算法和基本数据结构，讲解新标准下 C++ 的特性和使用方法，展现 C++ 发展的重要成果和编程风格。

本书借鉴了国内外同类教材的优点，在教学理念、教学内容和教学方法上有很大的改动。因此，本书有以下几个鲜明特点：

- 推行 C++11 新标准，紧随时代发展步伐。相比于 C++98 标准，C++11 新标准使得代码编写更简单、更高效，开发的程序性能也更好。本书教学内容的选定以 C++11 新标准为基础，强调程序的开发效率和性能。
- 突出编程思维和编程习惯的培养，提高编程素质。本书一方面从编译器的视角出发，来学习和理解 C++ 语法，从而更深层次地把握 C++ 的特性，为编写高质量的程序夯实基础。另一方面，本书也非常注重编程习惯的养成，比如代码的书写格式、对象和函数的命名、指针和引用的书写等。良好的编程习惯对于提高编程素质具有重要作用。
- 强调语言运用，提高工程实践能力。把 C++ 作为一种语言工具来学习，摒弃以语法为主的教学方法，通过一些常用算法和数据结构来展示如何利用 C++ 解决实际问题。
- 注重编程兴趣的培养。编程兴趣是学好编程语言的关键，为了激发读者对语言学习的渴望，本书在基础知识部分包含了许多有趣的小游戏，比如猜字谜、猜数字、石头剪刀布、扫雷等常见的游戏。
- 注重实际编程经验的介绍。本书通过警示框提醒读者容易出错的知识点，用提示框建议一些编程方法和编程习惯，帮助读者理解一些语法细节。

本书非常注重知识点的安排，从一开始就介绍面向对象编程和标准库中基本类对象的使用，强调对现代高级语言特性的理解。比如，本书强化对象意识，抛弃变量的概念，在介绍对象时就强化了对象是数据和操作的载体的意识。在讲解数组时，便介绍了标准库中 string 和 vector 的使用方法。这样做的好处是，避免低级语言特性先入为主，造成读者很难从面向过程的编程风格过渡到面向对象的编程风格的问题。

本书结构

本书的内容分为两部分。第 1 部分为基础篇，主要介绍 C++ 和标准库的基础内容。学完第 1 部分内容后，读者可以形成编程思维，养成良好的编程习惯，并能编写出具有一定意义的程序。

第 2 部分为设计篇，主要介绍基本数据结构、内存管理、面向对象程序设计方法、泛型编程和标准模板库。使用标准模板库能够提高代码开发效率，并保证代码质量；学习基本

数据结构能够帮助读者更好地理解和使用标准库中的容器与算法；优化内存管理将会进一步提升程序的性能。学完第 2 部分内容后，读者将进一步理解 C++ 的编程风格，能够利用面向对象和泛型技术编写大型软件。

教学安排

本书可以分为两个学期进行教学。第 1 学期完成基础篇教学，侧重于基本语法和常用算法，培养学生的编程兴趣，使学生具备将解题思路转化为代码的能力。第 2 学期完成设计篇教学，侧重于程序基本构架的设计及语言的运用，使学生具备一定的软件架构能力。

建议本书授课 72 学时，上机实验 32 学时：基础篇授课 40 学时、上机实验 16 学时；设计篇授课 32 学时、上机实验 16 学时。

致谢

本书内容由李长河主持编写，童恒建、叶亚琴和杨鸣参与相关内容的编写和审定，中国地质大学(武汉)王俊臣、刁义雅、肖龙、周力、刘永峰、夏海、黄祖传等博士和硕士研究生参与了本书内容的校对及课后习题的编写验证工作。

本书所有示例代码的源码及其他配套教学资源可从以下网址获得：

http://hxedu.com.cn 或 https://github.com/Changhe160/book-cplusplus。

感谢读者选择使用本书，欢迎您对本书内容提出意见和建议，我们将不胜感激。作者的电子邮件地址是：lichanghe@cug.edu.cn。

李长河
lichanghe@cug.edu.cn
于中国地质大学(武汉)

目录 Contents

第1部分 基 础 篇

第 2 部分 设 计 篇

① 作者注:“拷贝控制”为专业术语。

第 1 部分

基 础 篇

任何一门编程语言都有自己的语法(syntax)和语义(semantics)。语法规定了哪些符号的组合是正确的，语义是对程序的解释。因此，掌握一门编程语言的第一步就是要熟悉它的语法和语义，包括：

- 基本数据类型，如整型、字符型、实型等；
- 变量，用来存放某种类型的数据；
- 运算符和语句，用来操纵上述数据类型的值；
- 语句控制结构，用来操控语句的执行流程；
- 函数，可重复调用的执行某种特定功能的计算单元。

有了上面的基本元素，读者可以编写结构复杂、能解决实际问题的程序。C++ 不仅提供了基本数据类型，它还允许程序员定义自己的数据类型和相关的操作，这也是 C++ 强大的地方。这些用户自定义类型在 C++ 里称为类类型，类是 C++ 实现面向对象编程的基础，也是 C++ 的核心技术之一。

本书第 1 章 ~ 第 6 章为基础篇，讲解上述基本内容，包括基本数据类型和表达式、语句控制结构、复合数据类型、标准库中的 string 和 vector 类型、函数、类。学完基础篇内容之后，读者将掌握 C++ 的基本语法规则，积累一些经典算法，熟悉面向对象的编程风格，养成计算机编程思维和良好的编程习惯，并具备解决实际问题的能力。

第1章　初识C++程序

学习内容和目标

本章介绍C++程序的基本要素及程序的编译和调试。学习完本章内容之后，读者将可以：

- 掌握C++程序的基本组成、了解类的概念；
- 学会独立上机编写、调试及运行一个简单的C++程序。

1.1　编写一个简单的C++程序

计算机语言是人与计算机进行交互的工具，具有很强的工程实践性，所以学习一门新的语言最有效的办法就是坚持上机编写和调试代码。本章将编写求解圆柱体体积的简单程序。

每个C++程序都包含一个或者多个函数(function)，每个函数都是一个具有名字的代码块，其中有一个函数必须命名为main。操作系统通过调用main函数来运行C++程序，因此main函数也称为入口函数。代码清单1.1是一个简单的main函数的定义，它什么也不做，只是返回给操作系统一个整型值。

例1.1　一个空的main函数。

代码清单1.1　例1.1

```
1  int main(){        //程序从main函数开始执行
2      return 0;      /* 返回一个整型值 */
3  }
```

一个函数的定义由四部分组成：返回值类型(return type)、函数名(function name)、形参列表(parameter list)和函数体(function body)。其中，形参列表放在圆括号()内，函数体放在花括号{}内，两者都允许为空(函数的详细描述见5.1节)。代码清单1.1中，返回值类型为int(整型类型)，形参列表为空，函数体由一对花括号括起来的语句块(block of statements)组成。数据具有不同的类型，例如字符型数据'A'和整型数据1。int为整型类型数据，是C++的一种基本数据类型。

双斜线(//)为单行注释(comment)语句，以换行符结束，用来注解代码的功能或提示等信息，编译器将忽略双斜线右边所有的文本。另一种注释的方式是使用界定符(/* */)，以/*开始，到*/结束，注释的文本放到界定符之间，可以包含换行符，所以界定符

可以跨行，如代码清单 1.2 中标号为 3 的内容所示，每行都以 * 号开头用来显式标明注释语句。

代码清单 1.1 中，函数体只有一条 return 语句，该语句返回一个整型数 0，并以分号结束。C++ 语句通常以分号结束，每一条语句单独占用一行是一个好的习惯。

在代码清单 1.1 的基础上，编写求解圆柱体体积的简单程序。

例 1.2 已知圆柱体的底面半径和高分别是 6.5cm 和 12cm，求圆柱体体积。

代码清单 1.2 例 1.2

```
1  //单行注释：计算圆柱体体积程序
2  #include <iostream>
3  /*
   * 多行注释：C++标准输入输出流类库，
   * 用于两个 IO(输入和输出)对象 std::cin 和 std::cout
   */
4  int main() {
5      //定义三个 double 类型对象，存放半径、高和体积的值
6      double radius, height, volume;
7      //屏幕终端显示 Please input radius and height:
8      std::cout << "please input radius and height: ";
9      std::cin >> radius >> height;     //从键盘输入 6.5 12 后回车
10     //计算圆柱体体积，并把结果存放到对象 volume 中
11     volume = 3.14*radius*radius*height;
12     //屏幕输出 the volume is 1591.98
13     std::cout << "the volume is " << volume << std::endl;
14     return 0;
15 }
```

和代码清单 1.1 相比，代码清单 1.2 多了三部分的内容：(1)包含了输入输出(IO)流类库 iostream(第 2 行)；(2)定义了三个双精度(double)类型对象，即 radius、height 和 volume(第 6 行)，并进行了表达式计算(第 11 行)；(3)输入输出语句(第 8、9 和 13 行)。下面分别讲解这三部分代码。

大部分程序都需要对数据进行处理和计算，并给出计算结果，这就需要将数据存放到内存中，并能够读写内存中的数据。C++ 程序通过对象(object)来实现对内存数据的读写，例如代码清单 1.2 中定义了三个 double 类型的对象，用来存取 double 类型的数据，分别是圆柱体的底面半径 radius、高 height 和体积 volume。第 11 行代码计算圆柱体的体积，并把结果保存在对象 volume 中。

C++ 语言本身没有提供 IO 机制，而是通过标准库(standard library)中定义的输入输出流类库(iostream) 来实现 IO 操作。iostream 包含两个基础类型即 istream 和 ostream，分别为输入流和输出流，用来实现从字符序列到 IO 设备的写入和读出。代码清单 1.2 使用了输入流对象 cin 和输出流对象 cout，其中输入流对象 cin 负责向 radius 和 height 两个对象写入数据，输出流对象 cout 负责向用户屏幕输出相关信息。利用 cin 从流中读取数据，把数据保存到对象里面，需要使用输入运算符(>>)，如下：

```
std::cin >> radius >> height;
```

当从键盘输入 6.5 12 后回车时，这些字符会保存到与输入流对象 cin 相关联的缓冲区中，然后通过输入运算符 >>，可以将流中两个数据分别保存到 radius 和 height 对象里。类似 istream 对象 cin，利用输出运算符 << 将对象中保存的数据写到输出流中，如下：

```
std::cout << "the volume is " << volume << std::endl;
```

其中，"the volume is "是字符串字面值常量(string literal，见 2.2.2 节)，即用一对双引号引起来的字符序列，可以直接写入输出流中并在用户终端显示。cout 语句最后一个输入运算符后面的 endl 的功能是插入换行符并将缓冲区的内容刷新到终端设备。

要利用 cin 和 cout 实现 IO 操作，需要包含 iostream 库所在的头文件(header file)，通过 #include 指令告诉编译器需要使用 iostream 库。iostream 库名放到尖括号(< >)里面，而且必须把预编译指令 #include 和需要包含的头文件的名字放到同一行。

通常将 #include 指令放到文件的起始位置，如代码清单 1.2 所示。细心的读者可能会注意到：代码清单 1.2 中出现了 std，cin 和 cout 的前缀 std::表明 cin 和 cout 是定义在名为 std 的命名空间(namespace，见 12.1 节)中，其中::为作用域运算符，用来访问某个作用域(scope)下的名字。关于 IO 机制，将在第 10 章详细讲解。

1.2 认识类

类(class) 是 C++ 语言一个重要特性，是面向对象程序设计(Object-Oriented Programming，OOP)的一个基础。C++ 语言创始人 Bjarne Stroustrup 最初把 C++ 语言命名为“带类的 C(C with classes)”，这也体现了类在 C++ 中的地位。类的核心思想是定义一种数据结构(data structure)及与数据结构相关联的一组操作，并把它们封装在一起，形成一个类类型(class type)。

下面用面向对象的方法来求解例 1.2。

例 1.3 一个简单的圆柱体类。

代码清单 1.3 例 1.3

```
1  #include <iostream>
2  using  namespace std;      //使用标准命名空间
3  //将半径、高及两个操作函数封装在一起，形成一个类类型，命名为 Cylinder
4  class Cylinder {
5      double m_radius, m_height;
6  public:
7  //一个与半径和高相关联的操作，用来求体积并返回体积值
8      double volume() {
9          return 3.14*m_radius*m_radius*m_height;
10     }
11 //初始化半径和高的操作
```

```
12      Cylinder(double i=0, double h=0):m_radius(i), m_height(h){}
13  };
14
15  int main() {
16      Cylinder object(1.0, 1.0);          //定义并初始化对象 object
17      double vol=object.volume();         //调用 Cylinder 类中的操作 volume 函数
18      cout << vol << endl;
19      return 0;
20  }
```

在代码清单1.3中，除一个main函数外，利用关键字class定义了一个新的用户自定义类型，名字为Cylinder(第4行)，Cylinder类定义了一个含有半径(m_radius)和高(m_height)的数据结构，以及与之关联的两个操作。其中，一个操作为volume函数(第8行)，用来求圆柱体体积，另一个操作用来初始化半径和高(第12行)。这样把数据结构和两个操作函数用花括号{}封装起来并以分号结束(第13行)就构成了用户自定义类型的主体。这样做的一个好处是：类的使用者/用户不用关心类的具体实现，包括数据在内存中是如何存储的、操作是怎样实现的。这也是设计类的一个初衷。

在main函数中，语句

```
Cylinder object(1.0, 1.0);
```

定义了一个Cylinder类型的对象，对象的名字为object，它的两个数据成员均被初始化为1.0。这和代码清单1.2中定义一个double类型对象radius的方法是一样的，而且用户没有必要去知道怎样求解圆柱体体积，用户只需要知道Cylinder类提供了求解体积的操作和如何使用这个操作即可。要想使用Cylinder类中求体积的操作，用户只需要书写如下语句：

```
object.volume();
```

上面语句通过object对象调用volume操作。关于类的详细内容将在第6章中讲解。

提示：用户的概念

本书中所说的用户是类的使用者而非类的设计者，当然读者要学会如何为用户设计一个优秀的类。

细心的读者可能发现，代码清单1.3中第18行代码，cout没有前缀std::，这是因为在程序的开始处，通过如下语句告诉编译器将使用标准命名空间名字std(第2行)：

```
using namespace std;
```

这样，编译器如果遇到一个非用户或C++定义的标识符(identifier)，它会自动到使用的命名空间去查找。

1.3 编译与调试程序

当一个程序编写完毕之后，接下来需要编译它。程序的代码一般存放到一个或多个代码文件中，程序文件一般称为源文件(source file)。通常情况下，源文件的名字以.cpp 作为后缀名(.cc、.cxx 或.C 也可)。如何编译一个 C++ 程序，取决于操作系统和编译器。目前，大多数编译器都具有集成开发环境(Integrated Development Environment，IDE)，程序员可以方便地编写、调试和运行程序。这本书中所有的示例代码都在 Visual Studio 2015(VS2015)下编译和运行。VS2015 有着友好的开发和调试环境，而且实现了 C++11 核心语言规范中的绝大多数功能。

代码编写完成后，我们一般希望程序能够直接正确运行。但不幸的是，希望难以实现。通常我们需要花大量的时间和精力去调试代码中的问题。对于初学者，学会如何快速地调试和分析代码中的问题，对于学习 C++ 和提高编程能力是十分重要的。代码错误被称为臭虫(bug)，包括语法错误和逻辑(语义)错误。语法错误指在程序编译时出现的错误(也称为编译时错误)，逻辑错误指程序运行过程中出现的错误(也称为运行时错误)。首先需要改正语法错误，否则程序无法运行。当编译(compile)程序时，编译器会检查代码中的语法错误。根据 IDE 的提示，可以很容易地解决绝大多数语法错误。例如，在代码清单 1.2 的第 11 行语句中，如果写成：

```
volume = 3.14 * radius * radius * heihgt;
```

当编译程序(Ctrl + F7)时，编译器会提示如下错误：

```
errorC2065: 'heihgt': 未定义的标识符
```

用鼠标双击这条错误提示，就可以定位错误源，仔细观察发现对象名 height 被拼写成 heihgt，将其改正并重新编译即可。要想快速地解决语法错误，可以按顺序逐条地改正编译器产生的语法错误列表；有时对于某一个语法错误，编译器可能提示很多处其他的传递性错误，所以在修改完一个错误后立即重新编译，即遵循“编辑 - 编译 - 调试”的原则，可以大大提高语法错误的修改效率。

当改正完所有的语法错误之后，并不意味着程序可以正确运行，往往需要进行调试(debug)。这是因为编译器虽然能够找到语法错误，但是它不能发现程序中的逻辑错误。调试程序需要耐心和仔细分析，配合 IDE 提供的工具，根据程序的执行流程，逐行调试每一条语句，仔细分析和跟踪在每一条语句中对象值的变化情况，找到程序中的每一个 bug 并修正，直到程序正确运行为止。这是一个“痛”且快乐的过程，修改逻辑错误的过程也是编程思维不断改进的过程，所以自然会提高编程能力。编程思维的形成没有捷径可走，只有不断地编写和调试程序，在积累的过程中自然会形成编程思维。在 VS2015 中，调试过程中常用的几个快捷键有 F5、F7、F9、F10 和 F11，读者可查阅参考手册或者询问有经验的人来了解相关的使用细节。