

国外计算机科学教材系列

国家教育部  
双语教学示范课程  
使用教材

# 双语版 C++程序设计

[爱尔兰] Paul Kelly 著  
苏小红

Learn C++  
through English and Chinese



电子工业出版社  
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY  
<http://www.phei.com.cn>

国外计算机科学教材系列

# 双语版 C++ 程序设计

Learn C++ through English and Chinese

電子工業出版社

Publishing House of Electronics Industry

北京 · BEIJING

## 内 容 简 介

本书由在计算机程序设计方面有着丰富教学和实践经验的中外作者合作编写。共分 14 章内容,由浅入深全面介绍 C++ 程序设计方法。本书通俗易懂,例子贴近生活,尤其强调读者的亲自参与意识。所有实例经过精心挑选。每章都为初学者提供了常见错误分析,每章结尾有很多有趣的习题,可以提高读者上机编程的兴趣。

本书是国内首次出版的中英文对照混排式双语版 C++ 程序设计教材,既方便初学者熟悉相关概念和内容,也便于英文非母语的读者熟悉英文专业词汇。

本书可作为高等学校计算机相关专业或软件学院的 C++ 程序设计双语教材,也可供程序员和编程爱好者参考使用。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

### 图书在版编目(CIP)数据

双语版 C++ 程序设计 = Learn C++ through English and Chinese: 汉英对照 / (爱尔兰) 凯利 (Kelly, P.), 苏小红著. - 北京: 电子工业出版社, 2010.6

(国外计算机科学教材系列)

ISBN 978-7-121-10797-9

I. ①双… II. ①凯… ②苏… III. ①C 语言 - 程序设计 - 高等学校 - 教材 - 汉、英 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2010) 第 079445 号

策划编辑: 马 岚

责任编辑: 冯小贝

印 刷: 北京市顺义兴华印刷厂

装 订: 三河市双峰印刷装订有限公司

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 × 980 1/16 印张: 25.25 字数: 646 千字

印 次: 2010 年 6 月第 1 次印刷

印 数: 4000 册 定 价: 45.00 元

凡所购买电子工业出版社的图书有缺损问题, 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。

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

服务热线: (010) 88258888。

## 出版说明

21世纪初的5至10年是我国国民经济和社会发展的关键时期,也是信息产业快速发展的关键时期。在我国加入WTO后的今天,培养一支适应国际化竞争的一流IT人才队伍是我国高等教育的重要任务之一。信息科学和技术方面人才的优劣与多寡,是我国面对国际竞争时成败的关键因素。

当前,正值我国高等教育特别是信息科学领域的教育调整、变革的重大时期,为使我国教育体制与国际化接轨,有条件的高等院校正在为某些信息学科和技术课程使用国外优秀教材和优秀原版教材,以使我国在计算机教学上尽快赶上国际先进水平。

电子工业出版社秉承多年来引进国外优秀图书的经验,翻译出版了“国外计算机科学教材系列”丛书,这套教材覆盖学科范围广、领域宽、层次多,既有本科专业课程教材,也有研究生课程教材,以适应不同院系、不同专业、不同层次的师生对教材的需求,广大师生可自由选择和自由组合使用。这些教材涉及的学科方向包括网络与通信、操作系统、计算机组织与结构、算法与数据结构、数据库与信息处理、编程语言、图形图像与多媒体、软件工程等。同时,我们也适当引进了一些优秀英文原版教材,本着翻译版本和英文原版并重的原则,对重点图书既提供英文原版又提供相应的翻译版本。

在图书选题上,我们大都选择国外著名出版公司出版的高校教材,如Pearson Education培生教育出版集团、麦格劳-希尔教育出版集团、麻省理工学院出版社、剑桥大学出版社等。撰写教材的许多作者都是蜚声世界的教授、学者,如道格拉斯·科默(Douglas E. Comer)、威廉·斯托林斯(William Stallings)、哈维·戴特尔(Harvey M. Deitel)、尤利斯·布莱克(Uyless Black)等。

为确保教材的选题质量和翻译质量,我们约请了清华大学、北京大学、北京航空航天大学、复旦大学、上海交通大学、南京大学、浙江大学、哈尔滨工业大学、华中科技大学、西安交通大学、国防科学技术大学、解放军理工大学等著名高校的教授和骨干教师参与了本系列教材的选题、翻译和审校工作。他们中既有讲授同类教材的骨干教师、博士,也有积累了几十年教学经验的老教授和博士生导师。

在该系列教材的选题、翻译和编辑加工过程中,为提高教材质量,我们做了大量细致的工作,包括对所选教材进行全面论证;选择编辑时力求达到专业对口;对排版、印制质量进行严格把关。对于英文教材中出现的错误,我们通过与作者联络和网上下载勘误表等方式,逐一进行了修订。

此外,我们还将与国外著名出版公司合作,提供一些教材的教学支持资料,希望能为授课老师提供帮助。今后,我们将继续加强与各高校教师的密切联系,为广大师生引进更多的国外优秀教材和参考书,为我国计算机科学教学体系与国际教学体系的接轨做出努力。

电子工业出版社

## 教材出版委员会

- |    |     |   |
|----|-----|---|
| 主任 | 杨芙清 | 北京大学教授<br>中国科学院院士<br>北京大学信息与工程学部主任<br>北京大学软件工程研究所所长 |
| 委员 | 王 珊 | 中国人民大学信息学院教授<br>中国计算机学会副理事长，数据库专业委员会主任              |
|    | 胡道元 | 清华大学计算机科学与技术系教授<br>国际信息处理联合会通信系统中国代表                |
|    | 钟玉琢 | 清华大学计算机科学与技术系教授、博士生导师<br>清华大学深圳研究生院信息学部主任           |
|    | 谢希仁 | 中国人民解放军理工大学教授<br>全军网络技术研究中心主任、博士生导师                 |
|    | 尤晋元 | 上海交通大学计算机科学与工程系教授<br>上海分布计算技术中心主任                   |
|    | 施伯乐 | 上海国际数据库研究中心主任、复旦大学教授<br>中国计算机学会常务理事、上海市计算机学会理事长     |
|    | 邹 鹏 | 国防科学技术大学计算机学院教授、博士生导师<br>教育部计算机基础课程教学指导委员会副主任委员     |
|    | 张昆藏 | 青岛大学信息工程学院教授  |

# Preface

This textbook teaches the fundamentals of programming using C++, a programming language which supports the development of software using the object-oriented paradigm.

Although the book is primarily intended as a textbook for a programming module in a computer science course, it is equally suited to an individual familiar with another programming language and who now wants to learn how to program in C++.

This book focuses on core concepts and features of C++ while keeping the explanations as simple as possible. Drawing on their professional experience, the authors teach C++ programming largely by way of examples that are organised for easy step-by-step learning.

Like so many other programming languages, C++ contains numerous English technical terms that are difficult for all students, including the native English speaking student.

*Learn C++ through English and Chinese* explains C++ concepts and terminology in English with additional explanatory annotations in Chinese. This bi-lingual approach will be appreciated by Chinese students and will help them focus on the C++ language without being over-burdened by English technical terminology. Despite C++ being available on a wide variety of platforms, this book is not specific to any particular machine, compiler, or operating system. All programs are designed to be portable with little or no modification to a wide variety of platforms.

*Learn C++ through English and Chinese*

- Is a comprehensive introduction to programming in C++.
- Uses practical examples to explain difficult theoretical examples.
- Uses a step-by-step approach with detailed explanation of programming examples.
- Uses explanatory annotations written in Chinese.
- Provides end-of-chapter ‘programming pitfalls’ commonly experienced by learners.
- Provides a ‘quick syntax reference’ at the end of each chapter that summarises the C++ syntax covered in the chapter. This is a useful resource for experienced programmers as well as for learners.
- Provides end-of-chapter exercises, allowing the learner to test and re-enforce their understanding of C++.
- Is suitable for students new to programming and those familiar with some other language, such as C or Basic, and who now wish to learn C++.
- Is accompanied by a web site containing the example programs, solutions to selected exercises, frequently asked questions and links to other useful resources.

## Typographic Conventions

The line numbers to the left of the program examples are for reference purposes only and are not part of the C++ language.

When a new term is introduced it is in *italic* type.

C++ statements, keywords, program variables and values are in this font.

This font is used in examples to show values that should be typed at the keyboard by the user.

*Paul Kelly*  
*Dublin Institute of Technology*

# 前 言

针对国内大学逐渐与国际接轨的发展趋势，英语教学和双语教学逐渐被人们重视起来。一方面，在国家教育部的大力支持下，许多课程建设成了国家教育部双语教学示范课程，但是这些课程大多采用英文原版教材，而面向双语教学的双语版教材在国内实属罕见。另一方面，以“国际化、工业化”为办学理念，注重国际化、工业化人才培养的软件示范学院的部分课程还邀请了一些外籍教师进行全英语授课，但是由于目前国内学生的英语水平参差不齐，导致全英语授课的教学效果不是非常理想。本书正是在此背景和需求下应运而生的。

本书的第一作者是爱尔兰都柏林工业大学（DIT）的高级讲师 Paul Kelly。Kelly 老师长期从事程序设计类课程的教学工作，在程序设计类课程教学方面教学实践经验丰富，在国外已先后出版多本程序设计语言类书籍。自哈尔滨工业大学软件示范学院成立以来，一直作为外聘教师在哈尔滨工业大学软件示范学院从事程序设计方面的教学工作，对中国学生比较了解，针对其在教学中发现的问题，即初学者面临着既不熟悉专业术语和基本概念、又不熟悉英文专用词汇的双重困难，提出了出版英汉对照混排式双语版教材的思路，帮助学生在克服语言障碍的同时，能够更快更好地熟悉和掌握计算机程序设计方面的基础知识，为国内的双语教学提供了一种最佳的解决方案。

本书内容共分 14 章，由浅入深、全面介绍 C++ 程序设计方法，既适合于以 C++ 作为入门语言的读者，也适合于学习过其他程序设计语言后想再学习 C++ 的读者。本书的特点如下：

1. 使用非常实用和贴近生活的例子以及图示来通俗易懂地讲解难于理解的概念，介绍面向对象程序设计方法的同时，尤其强调读者的亲自参与意识。
2. 采用案例驱动和循序渐进方式，从一个应用实例出发，先利用现有知识编写出一个较为简单的程序，然后在此基础上，不断扩充，在扩充的过程中引入一个新的概念和知识点，逐渐编写出一个较大的程序，每个例程都有详细的讲解。有的章以一个例子为中心贯穿始终讲解，后面章节的例子还会重用已有的部分程序代码，前、后章节之间既有内容上的联系，也有例程上的联系。
3. 重点内容和段落给出了中文注解。
4. 每章后面都有一节介绍初学者编程时易犯的错误，以帮助初学者在程序设计中避免这些错误。
5. 每章后面都有快速语法参考，总结本章内容，便于读者快速查询相关内容。
6. 每章后面都有精心设计的、有趣的习题，便于读者测试和强化对相关内容的理解。
7. 有相关的教学网站（华信教育资源网，网址 <http://www.hxedu.com.cn>），方便读者下载示例的源代码、部分习题解答及教学课件等资料。

本书由在计算机程序设计方面有着丰富教学和实践经验的中外作者合作编写。本书是在国内首次出版中英文对照混排式双语版教材，适合低年级的学生对照阅读，既方便初学者熟悉相关



概念和内容，也便于母语不是英语的读者熟悉英文的专业词汇，尤其适用于作为双语教学示范课程的教材。

Paul Kelly 是一位治学非常严谨的教师，本书的第二作者苏小红在与他合著过程中，经常为一个细节内容的编写进行交流与讨论，书稿完成后又进行了多次校对工作。本着对所有读者负责的精神，我们真诚地欢迎读者对教材提出宝贵意见，可以通过发送电子邮件或在网站（<http://book.sunner.cn>）上留言等多种方式与我们交流讨论。作者E-mail地址为Paul.Kelly@comp.dit.ie, sxh@hit.edu.cn。

苏小红

哈尔滨工业大学计算机科学与技术学院

2010年5月

# 目 录

<b>Chapter One Introduction (绪论)</b> .....	1
1.1 What is a computer program? (什么是计算机程序?).....	1
1.2 Developing a computer program (开发计算机程序).....	2
1.2.1 Program development cycle .....	2
1.3 Learning C++ (学习 C++).....	4
1.4 Web site for this book (本书的网站).....	5
1.5 Brief history of C++ (C++ 简史).....	5
1.6 ANSI/ISO C++ standard (ANSI/ISO C++ 标准).....	5
<b>Chapter Two Beginning to Program in C++ (C++ 编程入门)</b> .....	6
2.1 Constants (常量).....	6
2.2 Variables (变量).....	6
2.3 Simple output to the screen (简单的屏幕输出).....	7
2.4 Comments (注释).....	9
2.5 Data types (数据类型).....	11
2.5.1 Short Integer data types .....	11
2.5.2 Long Integer data types .....	11
2.5.3 Boolean data types .....	11
2.5.4 Double Floating-point data types .....	11
2.6 Data type sizes (数据类型的大小).....	12
2.7 Operators (运算符).....	12
2.7.1 The assignment operator .....	12
2.7.2 Arithmetic operators .....	13
2.7.3 Increment and decrement operators .....	14
2.7.4 Combined assignment operators .....	17
2.8 Operator precedence (运算符的优先级).....	17
2.9 Type conversions and casts (类型转换和强转).....	19
Programming pitfalls .....	22
Quick syntax reference .....	24
Exercises .....	25
<b>Chapter Three Keyboard Input and Screen Output (键盘输入和屏幕输出)</b> .....	29
3.1 Simple keyboard input (简单的键盘输入).....	29

3.2 Manipulators (流操纵符).....	31
3.3 Single-character input and output (单个字符的输入和输出).....	34
Programming pitfalls .....	36
Quick syntax reference .....	36
Exercises .....	37
<b>Chapter Four Selection and Iteration (选择与循环)</b> .....	<b>38</b>
4.1 Selection (选择).....	38
4.1.1 The if statement .....	38
4.1.2 The if-else statement .....	39
4.1.3 Compound statements .....	40
4.1.4 Logical operators .....	41
4.1.5 Nested if statements .....	42
4.1.6 The switch statement .....	42
4.1.7 The conditional operator ? : .....	44
4.2 Iteration (循环).....	45
4.2.1 The while statement .....	45
4.2.2 The do-while loop .....	47
4.2.3 The for statement .....	48
4.2.4 Nested loops .....	50
Programming pitfalls .....	53
Quick syntax reference .....	55
Exercises .....	57
<b>Chapter Five Arrays and Structures (数组和结构体)</b> .....	<b>60</b>
5.1 Arrays (数组).....	60
5.1.1 Introduction .....	60
5.1.2 Initialising an array .....	64
5.1.3 Two-dimensional arrays .....	65
5.1.4 Initialising a two-dimensional array .....	67
5.1.5 Multi-dimensional arrays .....	68
5.2 Structures (结构体).....	69
5.2.1 Introduction .....	69
5.2.2 Declaring a structure .....	69
5.2.3 Initialising a structure variable .....	72
5.2.4 Nested structures .....	73
5.3 The typedef statement (typedef 语句).....	74
5.4 Arrays of structures (结构体数组).....	75
5.5 Enumerated data types (枚举数据类型).....	75

Programming pitfalls .....	77
Quick syntax reference .....	78
Exercises .....	78
<b>Chapter Six Strings (字符串)</b> .....	<b>82</b>
6.1 C-strings (C 风格字符串).....	82
6.2 C-string input and output (C 风格字符串的输入和输出).....	83
6.3 Accessing individual characters of a C-string (访问 C 风格字符串中的单个字符).....	88
6.4 C-string functions (C 风格字符串函数).....	88
6.4.1 Finding the length of a C-string.....	89
6.4.2 Copying a C-string .....	89
6.4.3 C-string concatenation .....	90
6.4.4 Comparing C-strings .....	90
6.4.5 Other C-string functions .....	91
6.4.6 Converting numeric C-strings to numbers .....	91
6.5 C++ strings (C++ 字符串).....	92
6.5.1 string initialisation and assignment .....	94
6.5.2 string concatenation .....	96
6.5.3 string length, string indexing and sub-strings.....	97
6.5.4 string replace, erase, insert and empty strings.....	99
6.5.5 string searching .....	100
6.5.6 string comparisons .....	102
6.5.7 string input .....	104
6.5.8 string conversions.....	105
6.6 Arrays of strings (string 类型的数组).....	107
6.7 Character classification (字符分类).....	107
Programming Pitfalls .....	110
Quick Syntax Reference .....	111
Exercises .....	111
<b>Chapter Seven Functions (函数)</b> .....	<b>114</b>
7.1 Introduction (引言).....	114
7.2 Function arguments (函数实参).....	116
7.3 Default parameter values (默认的形参值).....	120
7.4 Returning a value from a function (从函数返回一个值).....	120
7.5 Inline functions (内联函数).....	122
7.6 Passing arguments by value (按值传递实参).....	123
7.7 Passing arguments by reference (按引用传递实参).....	124
7.8 Passing a one-dimensional array to a function (向函数传递一维数组).....	127

7.9	Passing a multi-dimensional array to a function (向函数传递多维数组)	132
7.10	Passing a structure variable to a function (向函数传递结构体变量)	133
7.11	Passing a string to function (向函数传递字符串)	136
7.11.1	Passing a C++ string to a function	136
7.11.2	Passing a C-string to a function	137
7.12	Recursion (递归)	137
7.13	Function overloading (函数重载)	140
7.14	Storage classes auto and static (auto 和 static 存储类型)	142
7.14.1	auto	142
7.14.2	static	142
7.15	The scope of a variable (变量的作用域)	144
7.15.1	Block scope	144
7.15.2	Global scope	145
7.15.3	Reusing a variable name	146
7.16	Mathematical functions (数学函数)	148
7.16.1	Some trigonometric functions	148
7.16.2	Pseudo-random number functions	149
	Programming Pitfalls	151
	Quick Syntax Reference	151
	Exercises	152
<b>Chapter Eight</b>	<b>Objects and Classes (对象和类)</b>	<b>157</b>
8.1	What is an object? (什么是对象?)	157
8.2	What is a class? (什么是类?)	157
8.3	Further examples of classes and objects (类和对象的更进一步的示例)	160
8.3.1	A student class	160
8.3.2	A bank account class	161
8.4	Abstraction (抽象)	162
8.5	Constructing a class in C++ (在 C++ 中构造一个类)	162
8.6	Using a class: defining and using objects (使用类: 定义和使用对象)	165
8.7	Abstract data types (抽象数据类型)	166
8.8	Constructors (构造函数)	167
8.9	Default class constructor (默认类构造函数)	169
8.10	Overloading class constructors (重载类构造函数)	172
8.11	Constructor initialisation lists (构造函数初始化列表)	174
8.12	Default argument values in a constructor (构造函数中的默认实参值)	174
8.13	static class data members (静态类数据成员)	176
8.14	Using return in a member function (在成员函数中使用 return)	180

8.15	Inline class member functions (内联成员函数)	182
8.16	Class interface and class implementation (类的接口和类的实现)	183
8.16.1	Separation of class interface and class implementation	185
8.16.2	Use of namespaces in header files	187
	Programming Pitfalls	191
	Quick Syntax Reference	191
	Exercises	192
<b>Chapter Nine</b>	<b>Pointers and Dynamic Memory (指针和动态内存分配)</b>	<b>196</b>
9.1	Variable addresses (变量的地址)	196
9.2	Pointer variables (指针变量)	197
9.3	The dereference operator * (解引用运算符*)	198
9.4	Using const with pointers (使用 const 修饰指针变量)	199
9.5	Pointers and one-dimensional arrays (指针和一维数组)	201
9.6	Pointers and multi-dimensional arrays (指针和多维数组)	203
9.7	Pointers to structures (指向结构体的指针)	204
9.8	Pointers to class objects (指向类对象的指针)	206
9.9	Pointers as function arguments (指针变量作为函数实参)	207
9.10	Dynamic memory allocation (动态内存分配)	209
9.10.1	Allocating memory dynamically for an array	210
9.10.2	Initialisation with new	211
9.10.3	Allocating memory for multi-dimensional arrays	213
9.10.4	Out of memory error	215
	Programming pitfalls	217
	Quick syntax reference	218
	Exercises	219
<b>Chapter Ten</b>	<b>Operator Overloading (运算符重载)</b>	<b>222</b>
10.1	The need for operator overloading (运算符重载的必要性)	222
10.2	Overloading the addition operator + (重载加法运算符+)	223
10.3	Rules of operator overloading (运算符重载的规则)	230
10.4	Overloading ++ (重载运算符++)	231
10.4.1	Overloading prefix and postfix forms of ++	233
10.4.2	Improving the prefix ++ operator member function	237
10.5	Overloading relational operators (重载关系运算符)	237
10.6	Overloading << and >> (重载运算符<<和>>)	241
10.7	Conversion operators (转换运算符)	246
10.8	Use of friend functions (使用友元函数)	250
10.9	Overloading the assignment operator = (重载赋值运算符=)	251

10.9.1	A class with a pointer data member .....	251
10.9.2	Assigning one object to another.....	254
10.10	The copy constructor (拷贝构造函数) .....	263
10.11	Overloading the index operator [] (重载下标运算符[]) .....	268
	Programming pitfalls .....	272
	Quick syntax reference .....	273
	Exercises .....	273
<b>Chapter Eleven</b>	<b>Inheritance (继承)</b> .....	<b>276</b>
11.1	What is inheritance? (什么是继承?).....	276
11.2	Inheritance syntax (继承语法).....	278
11.3	Passing arguments to a base class constructor (向基类的构造函数传递实参).....	286
11.4	Protected class members (受保护的类成员).....	291
11.5	Types of inheritance: public, protected and private (继承的类型: public、protected 和 private).....	295
11.6	Composition (组合).....	296
11.7	Multiple inheritance (多重继承).....	298
11.8	Virtual base classes (虚基类).....	302
	Programming pitfalls .....	305
	Quick syntax reference .....	306
	Exercises .....	306
<b>Chapter Twelve</b>	<b>Polymorphism (多态性)</b> .....	<b>312</b>
12.1	What is polymorphism? (什么是多态?).....	312
12.2	Virtual functions (虚函数).....	316
12.2.1	When to use virtual functions .....	322
12.2.2	Overriding and overloading .....	322
12.3	Abstract base classes (抽象基类).....	322
	Programming pitfalls .....	327
	Quick syntax reference .....	328
	Exercises .....	328
<b>Chapter Thirteen</b>	<b>Templates (模板)</b> .....	<b>332</b>
13.1	Introduction (引言).....	332
13.2	Function templates (函数模板).....	332
13.3	Class templates (类模板).....	337
	Programming pitfalls .....	343
	Quick syntax reference .....	343
	Exercises .....	344

<b>Chapter Fourteen Files and Streams (文件和流)</b> .....	346
14.1 The C++ input/output class hierarchy (C++ 输入/输出类层次结构).....	346
14.2 Opening a file (打开文件).....	347
14.3 File error checking (文件出错检查).....	349
14.4 Single character I/O and detecting the end of a file (单字符的 I/O 和文件末尾 的检测).....	351
14.5 Appending data to the end of a file (向文件末尾添加数据).....	355
14.6 Reading lines from a file (从文件中读取行).....	356
14.7 Random access (随机存取).....	357
14.8 Object I/O (对象 I/O).....	361
14.9 Binary I/O (二进制 I/O).....	362
14.9.1 Serial writing of objects to a binary file .....	363
14.9.2 Serial reading of objects from a binary file .....	367
14.9.3 Binary I/O as class member functions .....	368
14.9.4 Binary file random access .....	369
Programming pitfalls .....	374
Quick syntax reference .....	374
Exercises .....	376
<b>Appendix A List of C++ Keywords</b> .....	378
<b>Appendix B Precedence and Associativity of C++ Operators</b> .....	379
<b>Appendix C ASCII Character Codes</b> .....	381
<b>Appendix D Fundamental C++ Built-in Data Types</b> .....	383
<b>Appendix E Common <code>io</code> Manipulators</b> .....	384
<b>Appendix F Escape Sequences</b> .....	385
<b>Appendix G The C++ Preprocessor</b> .....	386



# Chapter One

## Introduction

### 第 1 章 绪 论

#### 1.1 What is a computer program?

( 什么是计算机程序? )

Computers are involved in a wide variety of tasks that we do in our everyday lives. Some of these tasks such as using a word processor or checking e-mail obviously use a computer. Less direct examples occur when we use an ATM at a bank, pay at a supermarket checkout or use a phone.

A computer performs all of these tasks by following a predefined set of instructions. This set of instructions is called a *computer program*. A computer program to a computer is like a recipe to a chef; it specifies the steps needed to perform a certain task. But unfortunately, unlike a recipe, you can't give your instructions to a computer in a language such as English or Chinese. For instructions to be 'intelligible' to a computer, they need to be expressed in a language 'understood' by the computer. The only language 'understood' by a computer is its own machine language, which consists of a series of binary ones and zeroes.

Machine language is very difficult to use directly and so instructions to a computer are given in a special language called a *programming language*. The programming language is neither English nor machine language, but is somewhere in between. In fact, as you will see, it is more like English than machine language.

Machine languages are known as *low-level languages* and programming languages are known as *high-level languages*.

Writing instructions in a high level language is much easier than writing them in low-level machine language, but is still not as easy as writing them in English or Chinese.

现在我们日常生活中的很多工作都要用到计算机。

计算机完成所有这些工作都是通过执行预定义的指令集来实现的。这个指令集就称为“计算机程序”。

计算机程序和计算机之间,就像食谱和厨师之间的关系一样;计算机程序指定了完成某一任务需要的步骤。

但是遗憾的是,不同于菜谱,你不能用英文或者中文这样的自然语言向计算机发送指令。对于计算机来说,指令必须是“易理解”的,必须表示成一种计算机能“理解”的语言。计算机能“理解”的唯一语言就是机器语言,机器语言由一系列二进制的 0 和 1 组成。

由于机器语言很难直接使用,所以计算机指令被表示成一种特殊的语言,这种特殊的语言称为“程序设计语言”。程序设计语言既不是英语,也不是机器语言,而是一种介于它们两者之间的语言。实际上,正如下面即将看到的那样,较之机器语言,程序设计语言更像英语。

机器语言被认为是一种低级语言,而程序设计语言则被认为是一种高级语言。