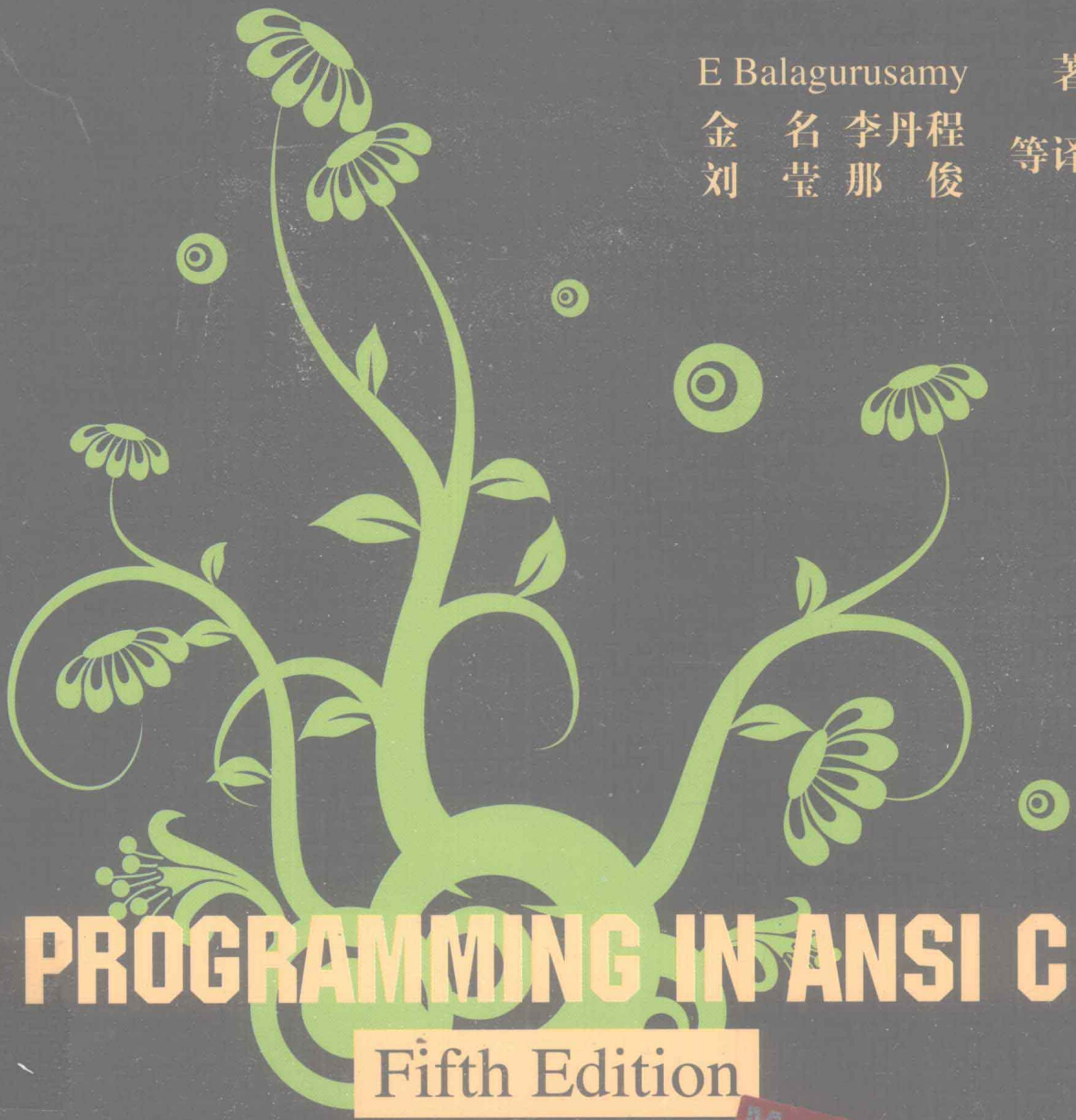
世界著名计算机教材精选

# 标准C程序设计（第5版）

E Balagurusamy 著
金 名 李丹程 等译
刘 莹 那 俊



# PROGRAMMING IN ANSI C

Fifth Edition

清华大学出版社

世界著名计算机教材精选

# 标准C程序设计

## （第5版）

E Balagurusamy 著

金 名 李丹程
刘 莹 那 俊 等译

清华大学出版社
北 京

E Balagurusamy
**Programming in ANSI C, 5e**
EISBN:0-07-068182-1



**图书在版编目(CIP)数据**

标准C程序设计（第5版）/（印）巴拉古路萨米（Balagurusamy, E.）著；金名，李丹程，刘莹，那俊等译. —北京：清华大学出版社，2011.9
（世界著名计算机教材精选）
书名原文：PROGRAMMING IN ANSI C, 5E
ISBN 978-7-302-26184-1

Ⅰ. ①标…　Ⅱ. ①巴…　②金…　③李…　Ⅲ. ①C 语言－程序设计－教材　Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字（2011）第 136974 号

**责任编辑**：龙啟铭
**责任校对**：白　蕾
**责任印制**：李红英

**出版发行**：清华大学出版社　　**地　　址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**邮　　编**：100084
**社　总　机**：010-62770175　　**邮　购**：010-62786544
**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
**印 装 者**：清华大学印刷厂
**经　销**：全国新华书店
**开　本**：185×260　　**印　张**：32.5　　**字　数**：785 千字
**版　次**：2011 年 9 月第 5 版　　**印　次**：2011 年 9 月第 1 次印刷
**印　数**：1～3000
**定　价**：59.00 元

产品编号：042471-01

# 译 者 序

C 语言是所有本科生课程的一门核心课，也是现今使用最广泛的计算机语言。国内国外的 C 语言图书已经非常多了，但通过本书的翻译，我们觉得这本书还是很有引进价值的，具体表现在它的以下几个特点上：

- 本书基于最新的 C 语言标准。
- 附录 IV 给出了 4 个完整的应用程序开发示例。
- 扩展讨论了 C 的指针。
- 每章后面的“谨记”一节给出了很有用的编程提示以及容易出错的问题。
- 每章后面的案例学习给出了 20 多个真实的开发，展示了 C 程序的设计过程。
- 80 多个程序设计范例向读者阐述了良好程序设计的基本原则。
- 还有近 200 个复习题和 130 多个项目设计题。

总之，本书的语言简洁易懂，示例非常丰富且具有很强的实际指导意义，是一本很好的 C 程序设计的教材。

本书的前面版本被印度的很多学院和大学用作教材。在过去的几年中，本书还被一些重要的软件培训与开发公司用作培训教材。

由于本书的独特性、与众不同的学习方法以及简明的写作风格，本书第 3 版和第 4 版在国内出版已受到国内学生和老师的欢迎。

本书由金名、李丹程、刘莹、那俊、石凯等主译，张长富、李晓春、王春桥、龚亚萍、韦笑、何雄、周云、袁科萍、王雷、贺军、贺民、陈安南、霍丽娜、史广飞、侯鹏、张红军、董武、陈河南、王峰、沈宏、郑晓蕊、李伟、白晓平、李月、汤效平、李东锋、邵世磊、张新苗、刘大为、薛飞、邹晓东、陈占军、夏绪虎、刘占坤、冯苗、裘蕾、任世华、金颖、吴霞、韩毅、马以辉、樊庆红等人也参与了部分翻译工作。欢迎广大读者指正。

# 第5版前言

C是功能强大、灵活性好、可移植且结构良好的程序设计语言。由于C集成了高级语言和汇编语言的优点，因此它可适用于系统和应用程序的开发。今天，C语言无疑是使用最广的通用语言。

自从1989年标准化以来，C经历了一系列的修订和改进，以提高该语言的可用性。现在，融合了这些新特性的版本称为C99。

# 本书结构

本书首先在第1章概要地介绍了C语言，包括C的历史、C程序的基本结构和运行。第2章讨论了如何声明常量、变量和数据类型。第3章讨论了内置操作符以及如何使用它们来构建表达式。第4章介绍了输入输出操作。第5章介绍了决策判断与分支，包括if-else、switch和goto语句。第6章讨论了决策与循环，包括while、do和for循环。第7章和第8章介绍了数组和数据元素的有序排列，这些对所有程序设计语言都很重要。第8章还介绍了字符串。第9章和第10章介绍了函数、结构和联合。指针是C语言中最难理解和掌握的部分了，第11章以对用户非常友好的方式介绍了指针。第12章和第13章分别介绍了文件管理和动态内存分配。第14章介绍了预处理器。最后，第15章开发了一个C程序，让读者了解如何进行程序的开发。如果读者按照本书结构认真学习，将可以很好地理解C语言。

本书内容进行了修订，以紧跟C程序设计领域的发展和当今的需要。与以往一样，全书仍然强调“用示例学习”的概念。在深入讨论了C语言的每个主要特性后，给出了一个完整的程序示例来演示其使用。示例程序既简单，又很有启发性。

每章的开头含有一小节，介绍本章的主要内容，并快速浏览了本章将要介绍的特色内容。只要需要，就使用图像来描述有关的概念，以提高描述的清晰性，方便读者理解。基本的语言技巧和其他需要特别考虑的内容以“注意”的形式凸显出来。为使本书更加实用，本书融合了以下主要特性。

- **带注释的代码**全书随处可见，这些注释说明了C语言的各种特性是如何集成在一起以实现特定任务的。
- **补充信息和“注意”**对正文进行了必要的补充，但独立于正文之外。
- 最后一章给出了开发高效的C程序的一些**指导原则**，以及经验不足的C程序员容易犯的常见错误列表。

- 每章末尾的**案例研究**演示了把C特性集成在一起的常用方式，并显示了一些实际的应用。
- 每章末尾的“**谨记**”一节列举了一些有用的提示和可能出问题的地方。
- 每章末尾的“**问题**”和“**练习**”为读者复习所学概念和实际的应用开发提供机会。
- 附录给出的**编程项目**介绍了开发大型程序时如何集成C的各种特性。

# 目　录

# 第1章 C语言概述

## 1.1 C语言的历史

对于一种程序设计语言，字母“C”看上去是一个奇怪的名字。但是这个奇怪而好听的语言却是现今最为流行的计算机语言之一，因为它是一种结构化的、高级的、与机器无关的语言。它使得软件开发人员在开发程序时，无须担心实现这些程序的硬件平台。

所有现代语言的起源都是 ALGOL 语言，该语言是在 20 世纪 60 年代早期引入的。ALGOL 是最先使用块结构的计算机语言。尽管它从来没有在美国流行开来，但在欧洲却被广泛使用。ALGOL 语言给计算机科学界带来了结构化程序设计的概念。20 世纪 60 年代，计算机科学家，如 Corrado Bohm、Guiseppe Jacopini 和 Edsger Dijkstra 使这一概念大众化了。随后，又宣布开发了好几种计算机语言。

1967 年，Martin Richards 开发了一种称为 BCPL（基本组合程序设计语言）的计算机语言。该语言主要用于系统软件的开发。1970 年，Ken Thompson 创建了一种计算机语言，该语言继承了 BCPL 的很多特性，就简单地称为 B 语言。在贝尔实验室，B 语言用来开发 UNIX 操作系统的早期版本。BCPL 和 B 语言都是“无类型”的系统程序设计语言。

C 语言是 Dennis Ritchie 于 1972 年在贝尔实验室从 ALGOL、BCPL 和 B 语言的基础上发展而来的。C 语言利用了这些语言的很多概念，并添加了数据类型的概念以及其他功能强大的特性。由于它是与 UNIX 操作系统一起被开发出来的，因此它与 UNIX 有着很强的关联。UNIX 操作系统（也是在贝尔实验室开发出来的）几乎完全是用 C 语言编码的。UNIX 是现今使用最为流行的网络操作系统，也是因特网数据超高速路的心脏。

多年以来，C 语言主要用于科研环境下，但后来，随着多种商用 C 编译器的发布，以及 UNIX 操作系统的不断流行，在计算机专业中也开始获得广泛支持。今天，C 语言可以运行在多种操作系统和硬件平台下。

20 世纪 70 年代，C 语言发展为现在所说的“传统 C 语言”。自 1978 年由 Brian Kerningham 和 Dennis Ritchie 著作的“The C Programming Language”一书的出版，C 语言成了最为流行的语言。该书很受欢迎，以至于在程序设计界，C 语言就认为是“K&R C”。C 语言的快速发展导致了不同版本的语言出现，这些语言相互类似但往往不兼容。这给系统开发人员带来了一个严重的问题。

为了确保 C 语言的标准，1983 年，美国国家标准局（American National Standards Institute，ANSI）任命了一个技术委员会来定义 C 语言的标准。该委员会于 1989 年批准了一个 C 语言版本，这就是现在的 ANSI C。该版本又于 1990 年被国际标准组织（International Standards Organization，ISO）批准。C 语言的这个版本又称为 C89。

20 世纪 90 年代，一种完全基于 C 的语言——C++语言，经历了大量的改进和变化，于 1989 年 11 月成为了一种获得 ANSI/ISO 批准的语言。C++在 C 的基础上添加了一些新特

性，使之不仅成为一种真正的面向对象的语言，而且是一种更通用的语言。与此同时，美国的Sun公司创造了一种新的语言——Java语言，它是以C和C++为模型的。

所有流行的计算机语言其本质都是动态的。它们通过加入新特性来不断地提高其功能和使用范围，C语言也不例外。尽管C++和Java语言都是从C发展而来的，C语言标准委员会认为，如果C++/Java语言的一些特性加入到C中，会提高C语言的性能。于是就产生了C语言的1999标准。这个版本的C语言通常称为C99。C语言的历史和发展如图1.1所示。

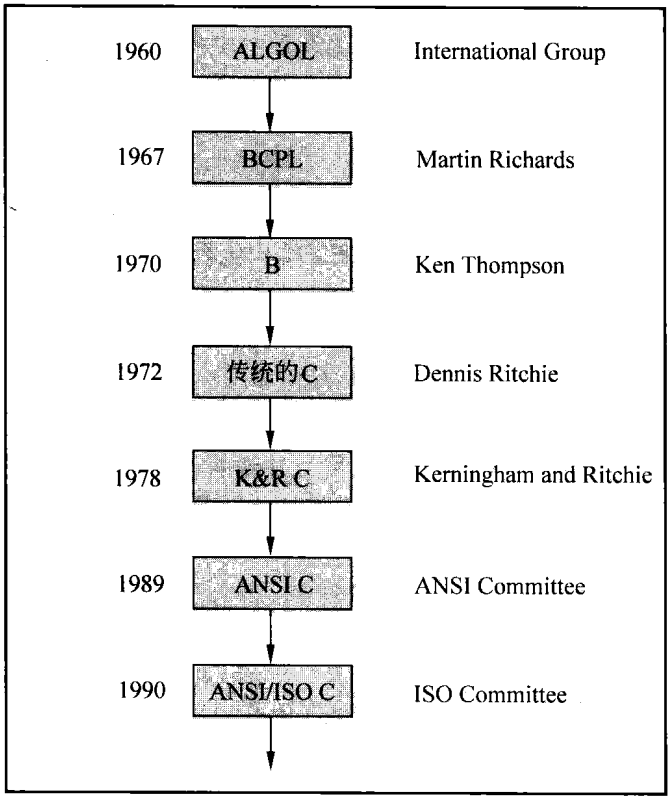



图1.1 C语言的历史

尽管C99是一个改进版本，但很多常用的编译器仍不支持C99的所有新特性。因此，我们将在附录中单独介绍加入到C99的新特性，这样，感兴趣的读者就可以快速地参考这些新特性，并在可能的情况下使用它们。

## 1.2 C语言的主要特性

C语言之所以越来越受欢迎，是因为它具有很多可取的特性。它是一种健壮的语言，其丰富的内置函数集和运算符可用来编写任意复杂的程序。C语言编译器融合了汇编语言的性能和高级语言的特性，因此很适合编写系统软件和商业软件。事实上，市场上很多的C编译器都是用C语言编写的。

用 C 语言编写的程序高效且运行速度快。这归功于它的各种数据类型和功能强大的运算符。其运行速度是 BASIC 语言的好几倍。例如，一个计算 0~15000 个变量的程序，用 C 语言编写，其运行时间为 1 秒，而用 BASIC 语言编写的则要运行 50 秒。

C 语言有 32 个关键字，它们的长度取决于其内置函数。C 语言有很多标准函数可用来开发程序。

C 语言是高度可移植的。这意味着，为某一台计算机编写的 C 程序，只要稍作修改甚至不用修改就可以在另一台计算机上运行。如果我们计划使用不同操作系统的新计算机，可移植性很重要。

C 语言很适于结构化程序设计，因而要求用户以功能模块的方式来思考问题。这些模块的恰当结合，就可以形成一个完整的程序。这种模块化的结构使得程序的调试、测试和维护更加容易。

C 语言另一个重要特性是它的自我扩展能力。一个 C 程序基本上是各种函数的集合，这些函数由 C 函数库支持。我们可以不断地将自己的函数添加到 C 函数库中去。由于有了这么大量的函数，程序设计就变得简单了。

在讨论 C语言的具体特征之前，我们先来看看一个 C 范例程序，分析并了解程序是如何工作的。

## 1.3　示例程序 1：显示一条消息

请看图 1.2 所示的一个很简单的程序。

```
main()
{
/*.........显示开始.........*/

    printf("I see, I remember");

    /*.......显示结束.........*/
}
```

图 1.2　显示一行文字的程序

该程序将产生如下的输出：

```
I see, I remember.
```

让我们来详细看看该程序。第一行告诉操作系统程序名是 main，从这一行开始执行。main()函数是 C 系统使用的一个特殊函数，用来告诉计算机程序的运行起点。每个程序必须正好有一个 main 函数。如果有多个 main 函数，编译器就无法知道哪一个是程序的起始点。紧跟在 main 后面的空括号对表明 main 函数不带参数。关于参数的概念我们将在讨论函数（第 9 章）时再详细介绍。

第二行的开始是“{”，表明 main 函数的开始，而最后一行的闭括号则表示该函数的结

束。在上面范例程序中，闭括号还标志着整个程序的结束。这两个括号之中的所有语句就形成了函数体。函数体包含有一个指令集，从而完成指定的任务。

在上面的示例程序中，函数体包含有3个语句，其中只有printf一行是可执行的语句。以/*开始，以*/结尾的行，称为注释行。程序中恰当地使用注释行，可以提高程序的可读性。更容易让人理解。由于注释行不是可执行语句，因此/*与*/之间的内容全部被编译器忽略掉。通常，一个注释可以插入到程序的任何空白处——行的开始、中间或结尾处——但不能插入到一个词的中间。

尽管注释行可以出现在程序的任意地方，但在C语言中它们不能嵌套。这就意味着，不能在注释行中再插入注释行。一旦编译器发现了注释的开始标志，它就将忽略掉后面的所有内容，直到再发现一个结束标志为止。下面注释行：

```
/* = = /* = = */ = = */
```

是不合法的，因此将产生一个错误。

由于注释行不会影响程序的运行速度以及编译后程序的大小，因此我们应大方地在程序中使用注释。注释有助于开发人员和其他用户理解程序的不同函数和运算，对程序的调试和测试也有帮助。我们将在后面的范例程序中体会到注释行的作用。

现在让我们来看printf( )函数，这是该范例程序中唯一可执行的语句：

```
printf("I see, I remember! ");
```

printf是预定义的标准的C函数，用于显示输出。预定义的含义就是，该函数已编写好并已编译。在链接时，与我们的程序链接在一起。编译和链接的概念将在本章的后面介绍。printf函数将两个引号之间的内容显示出来。在本示例程序中，其输出为：

```
I see, I remember!
```

注意，打印行以分号结尾。C语言的每条语句都必须以分号结尾。

假如我们要如下地将输出显示成两行：

```
I see,
I remember!
```

这可以通过添加两个printf函数来实现，具体如下：

```
printf("I see, \ n ");
printf("I remember! ");
```

两个括号之间的信息称为该函数的参数。第一个printf函数的参数是“I see, \ n”，第二个的是“I remember!”。这些参数只是要显示出来的字符串。

注意，第一个printf函数的参数在字符串的末尾包含有字符\和n的组合。该组合称为新行字符。新行字符命令计算机换到下一行（即新行）。在概念上，它类似于打字机的回车键。在显示了逗号字符后，新行字符\n 的出现将使“I remember!”显示在下一行。字符\和n之间不允许有空格。

如果省略掉第一个printf语句的新行字符，那么输出仍为一行，即如下所示：

```
I see,I remember!
```

这类似于图 1.2 中的程序的输出。但是，注意，在“,”和“I”之间没有空格。

只使用一个 printf 语句，只要在适当的地方使用新行字符，也可以生成两行或多行输出。例如，语句：

```
printf("I see, \n I remember! ");
```

的输出为：

```
I see,
I remember!
```

而语句：

```
printf("I \n.. see, \n… … … I\n… … … remember! ");
```

的输出为：

```
I
.. see,
… … … I
… … … remember!
```

注意，有些作者推荐在所有程序的开头，加上如下的输入输出库函数：

```
# include <stdio.h>
```

但是，对 printf 和 scanf 并没有必要，因为这两个函数已经定义为 C 语言的一部分了。关于输入输出函数的更多内容参见第 4 章。

在我们继续讨论其他更多的示例之前，应必须注意很重要的一点：C 语言是区分大小写字母的。例如，printf 和 PRINTF 并不相同。在 C 语言中，通常都是写成小写字母。而大写字母用作表示常量的符号名。我们也可以在输出字符串中使用大写字母，例如，“I SEE”和“I REMEMBER”。

上面介绍的用于显示“I see, I remember!”的示例是最简单的程序之一。图 1.3 列举了这类简单程序的一般格式。所有 C 程序都需要一个 main 函数。

**main 函数**

main 函数是每个 C 程序的一部分。C 语言允许有如下多种形式的 main 函数声明：

```
main()
int main()
void main()
main(void)
void main(void)
int main(void)
```

空括号对表示该函数不带参数。这也可以在括号中使用关键字 void 来明确表示不带参数。我们还可以在 main 之前指定 int 或 void。关键字 void 表示该函数不给操作系统返回任何信息，而 int 表示函数将返回一个整数值给操作系统。如果指定为 int，那么程序的最后一行必须是“return 0;”。

为了简单起见，我们在本书的所有示例程序中使用第一种形式的 main 函数。

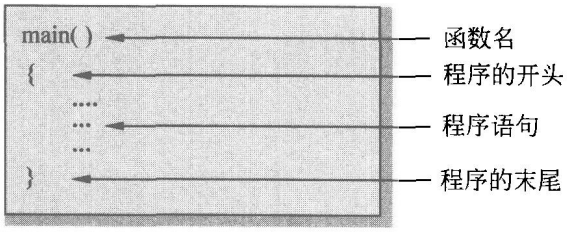



图 1.3 简单程序的结构

## 1.4 示例程序 2：两个数相加

下面来看另一个程序，它执行两个数的加法运算并显示其结果。整个程序如图 1.4 所示。

```
/* 程序名：ADDITION                          line-1  */
/* 作者：EBG                                 line-2  */
main()  /*                                   line-3  */
{                                        /*   line-4  */
    int   number;                        /*   line-5  */
    float amount;                        /*   line-6  */
                                         /*   line-7  */
    number = 100;                        /*   line-8  */
                                         /*   line-9  */
    amount = 30.75 + 75.35;              /*   line-10 */
    printf("%d\n",number);               /*   line-11 */
    printf("%5.2f",amount);              /*   line-12 */
}                                        /*   line-13 */
```

图 1.4 两个数相加的程序

当运行该程序时，将产生如下的输出：

```
100
106.10
```

该程序的头两行为注释行。在开始处添加注释行，用于给出诸如程序名、作者、编写日期等信息。这是一种很好的做法。其他行中也使用了注释符号，用于表示行号。

number 和 amount 字为变量名，用来存储数字数据。数字数据可以是整数型或实数型。在 C 语言中，所有变量都必须进行声明，从而告诉编译器所使用的变量名是什么、它们的数据类型是什么。变量必须在使用之前声明。在第 5 和第 6 行中，变量声明语句：

```
int number;
float amount;
```