

 四川工商职业技术学院
省级重点专业建设项目成果

程序设计基础

CHENGXU SHEJI
JICHU

主 编 朱凝秀 陆 岚 冯 云
副主编 杜若莲 黄 铭 赵定远（企业）



 西南交通大学出版社



四川工商职业技术学院

省级重点专业建设项目成果

程序设计基础

主 编	朱凝秀	陆 岚	冯 云
副主编	杜若莲	黄 铭	赵定远（企业）
参 编	陈 明	高晓琴	
主 审	潘 多		

西南交通大学出版社

· 成 都 ·

图书在版编目 (C I P) 数据

程序设计基础 / 朱凝秀, 陆岚, 冯云主编. —成都:
西南交通大学出版社, 2016.8
ISBN 978-7-5643-4867-0

I. ①程… II. ①朱… ②陆… ③冯… III. ①程序设
计 IV. ①TP311.1

中国版本图书馆 CIP 数据核字 (2016) 第 184905 号

程序设计基础

主编 朱凝秀 陆 岚 冯 云

责任编辑 穆 丰
封面设计 何东琳设计工作室

出版发行 西南交通大学出版社
(四川省成都市二环路北一段 111 号
西南交通大学创新大厦 21 楼)
发行部电话 028-87600564 028-87600533
邮政编码 610031
网 址 <http://www.xnjdcbs.com>

印 刷	四川煤田地质制图印刷厂
成 品 尺 寸	185 mm × 260 mm
印 张	9.25
字 数	230 千
版 次	2016 年 8 月第 1 版
印 次	2016 年 8 月第 1 次
书 号	ISBN 978-7-5643-4867-0
定 价	28.00 元

课件咨询电话: 028-87600533

图书如有印装质量问题 本社负责退换

版权所有 盗版必究 举报电话: 028-87600562

前 言

C 语言作为一门最通用的语言，在过去很流行，将来依然会如此。从 C 语言产生到现在，它已经成为最重要和最流行的编程语言之一。在各种流行编程语言中，都能看到 C 语言的影子，学习、掌握 C 语言是每一个计算机技术人员的基本功之一。

C 语言具有高级语言的强大功能，却又有很多直接操作计算机硬件的功能（这些都是汇编语言的功能）。学习和掌握 C 语言，既可以增进对于计算机底层工作机制的了解，又为进一步学习其他高级语言打下了坚实的基础。

C 语言的书林林总总，数不胜数，其中不乏《C Programming Language》这样的“Bible”级经典白皮书，也不乏《C 陷阱与缺陷》、《C Programming FAQs》这样的工程师枕边必备工具。中文教材方面也有《C 语言程序设计》这样广受学生们欢迎的教科书。

但是，对于刚开始接触编程的初学者来说，很难理解编程的思维，并且如何能够在编程语言的学习中得到阶段性的成就感——学习完某个知识点后，是不是可以实现一个什么功能？我们之所以要编写这样的一本教材，正是为了让初学编程的学生能够克服对编程的畏惧情绪，使他们能够在每一个知识点结束之后能够使用它去实现某个功能，使他们能够快速地实现知识到技能的转化。

我们采用了项目化的方式来编排这本书，把一个学生成绩管理系统分解成若干个独立的模块，在每个模块中分析相关的程序结构，引出相关的知识点。学生通过这些知识点的学习能很快弄清不同语句的功能，然后可以仿写出相似程序。到最后一章，再把各个小项目集合起来加以整合，就是一个完整的程序了。

同时，本书选取的主题，都是学习 C 语言必备的知识点。对每一个知识点都从正面解析清楚知识点的原理，然后通过程序来解释剖析，真正做到讲解精炼，基本上一两个实例即可说明关键知识点。本书阅读和学习门槛相对较低，读者阅读和学习的时候不会觉得枯燥。

我们希望这本教材能给初入门的程序设计学习者带来一定帮助，能激起他们学习编程的兴趣，能给他们继续前行的动力。

鉴于编者水平有限，书中不妥之处在所难免，恳请读者批评指正！

编 者

2016 年 6 月

目 录

阶段一 走进程序的世界	1
一、案例展示：学生成绩管理系统	1
二、案例解析	2
(一) 任务描述	2
(二) 任务目标	2
(三) 知识链接	2
三、作品实操	6
(一) 任务描述	6
(二) 效果展示	7
(三) 考核要点	7
四、知识拓展	7
(一) 算法的特征	7
课外阅读	9
阶段二 学生管理系统的欢迎界面	11
一、案例展示：登录界面与加法器	11
二、案例解析	13
(一) 任务描述	13
(二) 任务目标	14
(三) 知识链接	14
(四) 技术分析	20
三、作品实操	21
(一) 任务描述	21
(二) 效果展示	22
(三) 考核要点	22
四、知识拓展	22
(一) 关键字和标识符	22
(二) 定义常量	23
(三) 基本数据类型、整型变量、浮点型常量	24
(四) 运算符优先级	24
课外阅读	27

阶段三 学生管理系统中单个学生成绩的处理	29
一、案例展示：单个学生成绩的录入、统计和等级转换	29
二、案例解析	31
(一) 任务描述	31
(二) 任务目标	31
(三) 知识链接	31
(四) 技术分析	35
三、作品实操	38
(一) 任务描述	38
(二) 效果展示	38
(三) 考核要点	38
四、知识拓展	39
(一) 选择结构的其他实现	39
(二) 循环结构的其他实现	42
课外阅读	49
阶段四 学生管理系统中多个学生成绩的处理	52
一、案例展示：成绩的批量录入、排序以及学生成绩情况的输出	52
二、案例解析	53
(一) 任务描述	53
(二) 任务目标	53
(三) 知识链接	54
(四) 技术分析	58
三、作品实操	63
(一) 任务描述	63
(二) 效果展示	64
(三) 考核要点	64
四、知识拓展	64
(一) 多维数组	64
课外阅读	70
阶段五 学生管理系统中各功能模块的实现	72
一、案例展示：使用函数实现学生成绩管理系统的各种功能	72
二、案例解析	74
(一) 任务描述	74
(二) 任务目标	74
(三) 知识链接	75
(四) 技术分析	81

三、作品实操	87
(一) 任务描述	87
(二) 效果展示	87
(三) 考核要点	88
四、知识拓展	88
(一) 结构体知识拓展	88
(二) 函数知识拓展	92
课外阅读	98
阶段六 学生管理系统中数据的存取	101
一、案例展示：数据在文件中的存取	101
二、案例解析	103
(一) 任务描述	103
(二) 任务目标	103
(三) 知识链接	104
(四) 技术分析	109
三、作品实操	114
(一) 任务描述	114
(二) 效果展示	115
(三) 考核要点	116
四、知识拓展	116
(一) 文件指针	116
(二) 打开关闭文件	116
(三) 操作文件	118
课外阅读	123
阶段七 学生管理系统的软件开发	125
一、作品实操	125
(一) 任务描述	127
(二) 效果展示	127
(三) 考核要点	136
二、知识拓展	136
(一) 软件开发需求分析	137
(二) 软件质量保证——编码规范	137
(三) 软件测试	137
课外阅读	138
参考文献	140

阶段一 走进程序的世界

1997 年的 5 月,由 IBM 公司开发出来的“深蓝”计算机,战胜了当时的国际象棋世界冠军卡斯帕罗夫。卡斯帕罗夫是俄罗斯人,曾经被认为是有史以来最厉害的棋手之一,但他却被计算机打败了。2016 年的 3 月,由谷歌开发的 AlphaGo 与世界顶级围棋高手李世石对战,在五番棋“人机大战”当中,以 4:1 的结果完胜李世石。大家要知道,围棋的棋盘有 19×19 个格子,361 个点,每个点有 3 种可能,即黑棋、白棋或空白,这样,棋盘的每一个状态就有 3^{361} 种可能,这是一个天文数字,然而随着人工智能的快速发展,让原本人们认为计算机在短时间内难以攻克的围棋游戏也被突破。计算机为什么能完成那么多不可思议的任务呢?原因主要有两个:一是计算机的运算速度非常快;二是计算机的精度非常高,它可以不知疲倦地计算,而且不会出错。例如,根据一项统计数据,计算机在访问磁盘的时候,每隔 10 亿个数据位才可能会发生一个错误。所以说,计算机的工作原理虽然很简单,但由于它速度快、精度高,所以能够实现很多不可思议的功能。

虽然计算机的功能非常强大,但需要指出的是,计算机并不能直接帮助人们去解决问题。因为人们在描述这些问题的时候,采用的都是人类的自然语言的形式,而不是机器指令的形式,这样的话,计算机就“听不懂”人们所说的话,更不知道该去做什么。那么,我们要怎么做才能够让计算机来帮助我们解决这个问题呢?答案就是计算机程序!

一、案例展示:学生成绩管理系统

图 1-1 所示是学完本书后我们可以实现的学生成绩管理系统的主界面,这个程序是由 C 语言编写的。

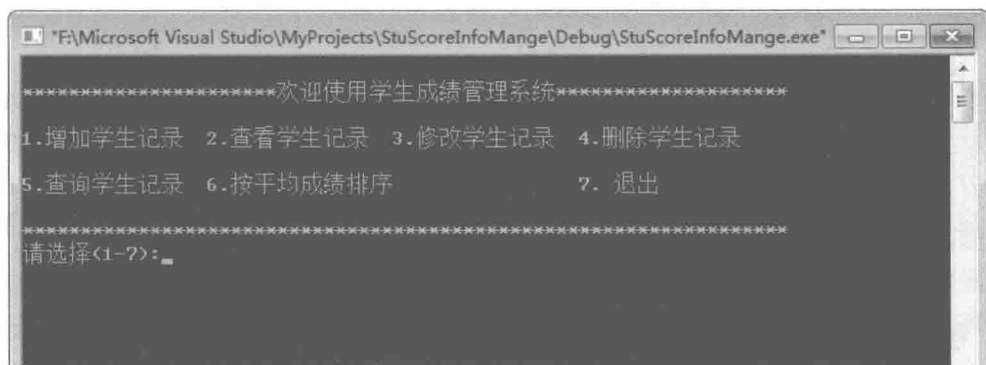


图 1-1 成绩管理系统主界面

学生成绩管理系统具有 7 大功能,分别是增加学生记录、查看学生记录、修改学生记录、

删除学生记录、按姓名查询学生记录、按平均成绩排序以及退出系统。该案例向用户展现了一个菜单选择界面，用户可以根据菜单界面的提示，选择不同的功能进入子界面。

通过上述功能描述，我们管理学生的成绩将变得简单且方便。要实现上述功能，我们就必须学会程序设计的方法。在本书的后续章节我们将分步骤给大家讲解。接下来我们来看一个更加简单的案例。

二、案例解析

（一）任务描述

为了弄清楚什么是程序，以及人们是如何编写程序的，我们先学习下面简单的例子。

例 1.1 用程序实现求 $1+2+3+\cdots+100$ 的和。

求和本是一个很简单的案例，结果是 5050 想必大家在小学就知道了，但是我们要让计算机替我们去计算，就需要同学们设计好算法，并用某种程序语言实现它。为此我们首先要搞明白以下几个问题：

- （1）什么是计算机程序？
- （2）什么是算法？
- （3）算法是如何表示的？

（二）任务目标

要完成本案例，我们需要掌握以下几点：

- （1）了解程序的概念；
- （2）掌握算法的概念；
- （3）掌握算法的表示。

（三）知识链接

1. 计算机程序

程序设计（Programming）是给出解决特定问题的程序的过程，是软件构造活动中的重要组成部分。程序设计往往以某种程序设计语言为工具，给出这种语言描述下的程序。程序设计过程应当包括分析、设计、编码、测试、排错等不同阶段。专业的程序设计人员常被称为程序员。

2. 程序设计的步骤

（1）分析问题：对于接受的任务要进行认真的分析，研究所给定的条件，分析最后应达到的目标，找出解决问题的规律，选择解题的方法，完成实际问题。

(2) 设计算法：即设计出解题的方法和具体步骤。

(3) 编写程序：根据得到的算法，用一种高级语言编写出源程序并通过测试。然后对源程序进行编辑、编译和连接。

(4) 运行程序，分析结果：运行可执行程序，得到运行结果。能得到运行结果并不意味着程序正确，要对结果进行分析，看它是否合理。结果不合理的，要对程序进行调试，即通过上机发现和排除程序中的故障。

(5) 编写程序文档：许多程序是提供给别人使用的，如同正式的产品应当提供产品说明书一样，正式提供给用户使用的程序，必须向用户提供程序说明书。内容应包括：程序名称、程序功能、运行环境、程序的装入和启动、需要输入的数据以及使用注意事项等。

3. 程序设计的语言

(1) 释义：就像人与人之间的交流，计算机的交流一样需要语言（计算机程序），只有这样才能执行命令。

(2) 具体释义：是用于编写计算机程序的语言。语言的基础是一组记号和一组规则，根据规则由记号构成的记号串的总体就是语言。在程序设计语言中，这些记号串就是程序。程序设计语言包含三个方面，即语法、语义和语用。语法表示程序的结构或形式，亦即表示构成程序的各个记号之间的组合规则，但不涉及这些记号的特定含义，也不涉及使用者；语义表示程序的含义，亦即表示按照各种方法所表示的各个记号的特定含义，但也不涉及使用者；语用表示程序与使用的关系。

(3) 语言的类型：ActionScript、APL、A+、B 语言、J 语言、Ada、汇编语言、AWK、Basic、Fortran、VBScript、C 语言、JAVA 语言、C#语言……总之，程序设计的语言有很多。

(4) 规范：程序设计的基本概念有程序、数据、子程序、子例程、协同例程、模块以及顺序性、并发性、并行性和分布性等。程序是程序设计中最为基本的概念。子程序和协同例程都是为了便于进行程序设计而建立的程序设计基本单位，顺序性、并发性、并行性和分布性反映程序的内在特性。程序设计是软件开发工作的重要部分，而软件开发是工程性的工作，所以要有规范。程序设计规范是进行程序设计的具体规定。语言影响程序设计的功效以及软件的可靠性、易读性和易维护性。专用程序为软件人员提供合适的环境，便于进行程序设计工作。

4. 算法及算法的表示

算法（algorithm）一词源于算术（algorism），即算术方法，是指一个由已知推求未知的运算过程。后来，人们把它推广到一般领域，把进行某一工作的方法和步骤称为算法。算法又可以理解为有基本运算及规定的运算顺序所构成的完整的解题步骤，它是求解问题类的、机械的、统一的方法。它由有限多个步骤组成，对于问题类中的每个给定的具体问题，机械地执行这些步骤就可以得到问题的解答。算法也可以看成按照要求设计好的有限的确切的计算序列，并且这样的步骤和序列可以解决一类问题。用计算机解决问题的过程可以分成三个阶段：分析问题、设计算法和实现算法。

或者看成按照要求设计好的有限的确切的计算序列，并且这样的步骤和序列可以解决一

类问题。用计算机解决问题的过程可以分成三个阶段：分析问题、设计算法和实现算法。

我们已经知道了算法的概念，那么我们在现实应用中怎样表示一个算法呢？除了很简单的问题，一般不用自然语言表示算法。通常我们会用流程图、伪代码、计算机语言表示算法。

1) 流程图

用流程图表示算法，直观形象，易于理解。流程图所包含的元素如图 1-2 所示。

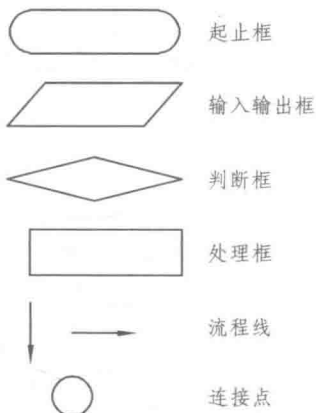


图 1-2 流程图的元素

一个流程图包括：表示相应操作的框；带箭头的流程线；框内外必要的文字说明。

将例 1.1 用流程图表示，求 $1+2+3+\cdots+100$ 的和的算法，如图 1-3 所示。

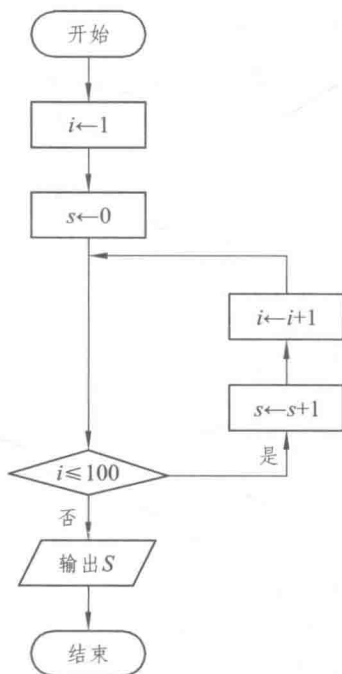


图 1-3 流程图

(1) 三种基本结构。

① 顺序结构如图 1-4 所示。

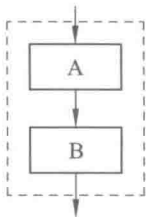


图 1-4 顺序结构流程图

② 选择结构如图 1-5 所示。

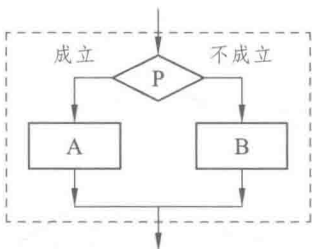


图 1-5 选择结构流程图

③ 循环结构如图 1-6 所示。

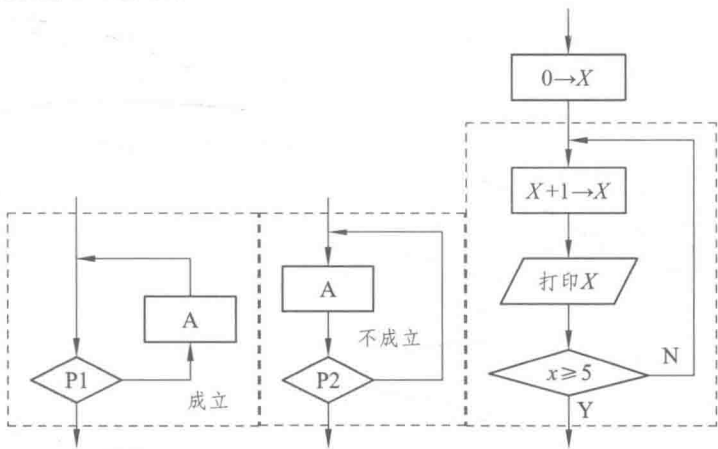


图 1-6 循环结构流程图

三种基本结构的共同特点：只有一个入口；只有一个出口；结构内的每一部分都有机会被执行到；结构内不存在“死循环”。

(2) 改进的流程图。

1973 年美国学者提出了一种新型流程图：N-S 流程图。

① 顺序结构如图 1-7 所示。



图 1-7 顺序结构

② 选择结构如图 1-8 所示。

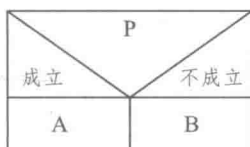


图 1-8 选择结构

③ 循环结构如图 1-9 所示。

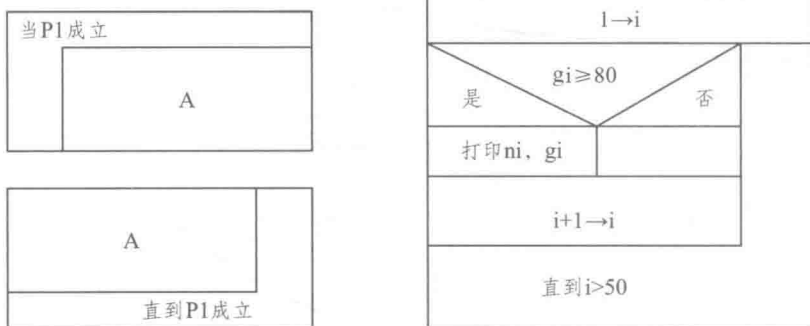


图 1-9 循环结构

2) 伪代码

伪代码使用介于自然语言和计算机语言之间的文字和符号来描述算法。

3) 用计算机语言表示算法

我们的任务是用计算机解题，就是用计算机实现算法，用计算机语言表示算法必须严格遵循所用语言的语法规则。

将例 1.1 用 C 语言表示。

```
void main()
{
    inti,sum=0;
    for(i=1;i<=100;i++)
        sum=sum+i;
    printf("%d",sum);
}
```

三、作品实操

(一) 任务描述

例 1.2 求 $1+2+3+\dots+100$ 的和。

最原始方法：

步骤 1: 先求 $1+2$, 得到结果 3。

步骤 2: 将步骤 1 得到的和 3 加上 3, 得到结果 6。

步骤 3: 将 6 再加上 4, 得 10。

.....

步骤 100: 将 4950 再加上 100, 得 5050。

这样的算法虽然正确, 但很繁琐。

改进的算法:

S1: 使 $i=1$ 。

S2: 使 $s=0$ 。

S3: 使 $s+i$, 和仍然放在在变量 s 中, 可表示为 $s+i \rightarrow s$ 。

S4: 使 i 的值+1, 即 $i+1 \rightarrow i$ 。

S5: 如果 $i \leq 100$, 返回重新执行步骤 S3 以及其后的 S4 和 S5; 否则, 算法结束。

该算法不仅正确, 而且是对计算机较好的算法, 因为计算机是高速运算的自动机器, 实现循环轻而易举。

(二) 效果展示

以下是我们求 $1+2+3+\cdots+100$ 的和的程序及运行结果, 如图 1-10 所示。

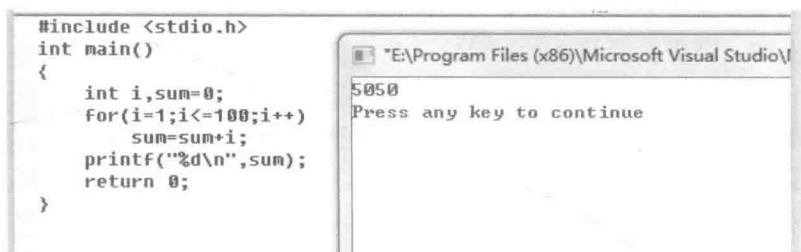


图 1-10 程序运行效果图

(三) 考核要点

(1) 程序的概念。

(2) 算法的表示。

四、知识拓展

(一) 算法的特征

上面的算法及算法的表示小节中已经学习了算法的基本概念以及算法的表示方法。下面我们一起来了解一个算法应该具有的五个方面的重要特征:

(1) 输入：所谓输入是指算法在执行时需要从外界取得的必要的信息。一个算法有零个或多个输入，以刻画运算对象的初始情况。

(2) 确定性：算法的每一个步骤必须要确切地定义。即算法中所有有待执行的动作必须严格而不含混地进行规定，不能有歧义性，以保证算法的实际执行结果精确地符合要求或期望，通常要求实际运行结果是确定的。

(3) 有穷性（有限性）：一个算法在执行有限步之后必须结束。也就是说，一个算法，它所包含的计算步骤应是有限的。

(4) 输出：算法有一个或多个输出，与输入有某个特定关系的量，简单地说就是算法的最终结果。

(5) 有效性（可行性）：算法中有待执行的运算和操作必须是相当基本的，换言之，它们都是能够精确地进行的，算法执行者甚至不需要掌握算法的含义即可根据该算法的每一步骤要求进行操作，并最终得出正确的结果。

算法中指令描述的是一个计算，当其运行时能从一个初始状态（可能为空的）和初始输入开始，经过一系列有限而清晰定义的状态，最终产生输出并停止于一个终态。一个状态到另一个状态的转移不一定是确定的。随机化算法在内的一些算法，包含了一些随机输入。

如何学习和使用程序设计语言

作为初学者,如何学会、掌握一门乃至多门程序设计语言?如何用所学的程序设计语言去解决实际工作中的应用问题呢?如果不考虑应用问题涉及的一些具体数据结构以及相应算法,则主要涉及三个方面的学习和训练。第一,要学习并理解所学程序设计语言的语法和语义;第二,要学习并熟悉与该程序设计语言对应的集成开发环境(Integrated Development Environment,简称IDE);第三,要熟悉与该程序设计语言相关的库函数(如C提供的各种各样的标准库函数)或类库(如Microsoft Visual C++提供的类库MFC)。

1. 理解程序设计语言的语法和语义

理解程序设计语言的语法和语义是学习程序设计语言的第一步。语法和语义原本都是数理逻辑名词。语义指的是形式系统中的符号、公式、公理、定理等的解释和意义。语法指的是对于形式系统的描述和研究中,对涉及公式的形式结构和定理的形式证明。

在程序设计语言中,语义指的是构成该语言的各种符号的含义。由于程序是由程序设计语言的符号所组成,因此只有准确理解各种符号的语义,才有可能合适地使用各种符号来表达自己的编程思想。另一方面,程序设计语言的语法规定了语言符号集中各种符号的结合方式,规定了各种表达式、语句、过程、函数乃至程序结构等的构造规则。因此,只有理解和掌握程序设计语言的语法,才有可能正确地用各种符号去构造程序的各种成分,并进而构成满足算法要求、能解决实际应用问题的程序。总之,程序不是符号的随意堆砌,而是各种符号根据其语义按一定的语法规律的有机构成。所以,只有理解和掌握了程序设计语言的语法和语义,才有可能正确地进行程序设计,才有可能读懂别人编写的程序。

在学习程序设计语言过程中,对程序设计语言的语法和语义的掌握程度根据初学者的专业和学习目标,也有不同程度的要求。对于仅仅要求学会使用程序设计语言的人而言,对语法和语义的掌握只要做到知其然,也就是能用即可。即做到能够正确地进行程序设计,能够读懂程序即可。对那些不仅希望知其然,还希望知其所以然的读者,尤其是对于计算机专业的学生而言,则不仅要会用,而且还要学会语言的形式化描述,学会分析语法成分的合理性、类型的抽象性等。因为,对他们而言,不仅要学习和掌握程序设计语言,而且要为有朝一日设计和创造新的程序设计语言奠定坚实的语言基础。

2. 熟悉集成开发环境

早期的编程者都是在行编辑程序(如EDLIN)或者全屏幕编辑程序(如WORDSTAR、SEE等)下编辑创建源程序,也称为源文件(如.c程序)。然后以命令行方式对源程序进行编译,产生对应的目标程序(如.obj文件)。由于目标程序地址的浮动性,它还需要经过链接程序Link进行链接,生成对应的可执行程序(如.exe文件)。最后将可执行程序投入运行以得到运行结果。

时至今日,几乎所有的程序设计语言都有自己的集成开发环境。集成开发环境将源程序

的创建、编辑以及其后的编译、链接，乃至程序的调试与跟踪全部都集成与一体。为编程者提供了方便的程序开发环境。因此，编程者要能够进行程序的设计和开发，调试与跟踪，就必须学会使用相应的集成开发环境。

集成开发环境一般都是 GUI 界面，以菜单和图标的方式提供各种操作功能。如：在 VC 的集成开发环境下，有 File、Edit、Run 等菜单板，每个菜单板下又有不同的菜单项。如 File 下又有 New、Open、Save 等菜单项。编程者只有学会使用集成开发环境提供的这些功能，才能进行程序的设计与开发。

3. 熟悉库函数或类库

对于多数程序设计语言，其编译系统（或者解释系统）一般都会提供一些标准的库函数。如输入/输出的标准函数、字符串处理的标准函数、数学计算的标准函数等。编程者只要知道这些标准函数的功能、调用方式、返回何种结果，并且能够在自己的程序中调用这些标准函数就行了。完全没有必要自己去编写这些函数。

对于面向对象的程序设计语言，其编译系统一般都会提供一些标准的类库。在这些类库中，封装了各种各样的数据成员和成员函数。编程者只要了解这些数据成员的含义，以及这些成员函数的功能，函数原型，就可以通过继承的方式使用这些现成的数据和函数。