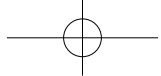


目 录

第1章 C语言基础知识	1
1.1 C语言简介	2
1.2 程序设计的基本概念	4
1.3 C语言程序的组成	9
1.4 程序的开发过程及环境	12
本章小结	15
自测题	15
实验1 熟悉Visual C++6.0集成开发环境	18
第2章 数据类型、运算符和表达式	20
2.1 数据表示	21
2.2 常量与变量	24
2.3 运算符及其表达式	27
本章小结	34
自测题	34
实验2 数据类型和表达式	37
第3章 顺序结构程序设计	39
3.1 字符的输入与输出	40
3.2 格式化输入与输出	42
3.3 顺序结构程序设计举例	47
本章小结	51
自测题	51
实验3 顺序结构程序设计	54
第4章 选择结构程序设计	57
4.1 if语句	58
4.2 条件运算符和条件表达式	66
4.3 switch语句	69
4.4 选择结构程序设计举例	71
本章小结	75
自测题	75
实验4 选择结构程序设计	80



第5章 循环结构程序设计 85

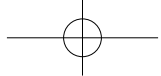
5.1 while 循环	86
5.2 do...while 循环.....	88
5.3 for循环及三种循环结构比较	91
5.4 break语句和continue语句	94
5.5 循环的嵌套	98
5.6 循环结构程序举例	101
本章小结	103
自测题.....	104
实验5 循环结构程序设计	111

第6章 数组117

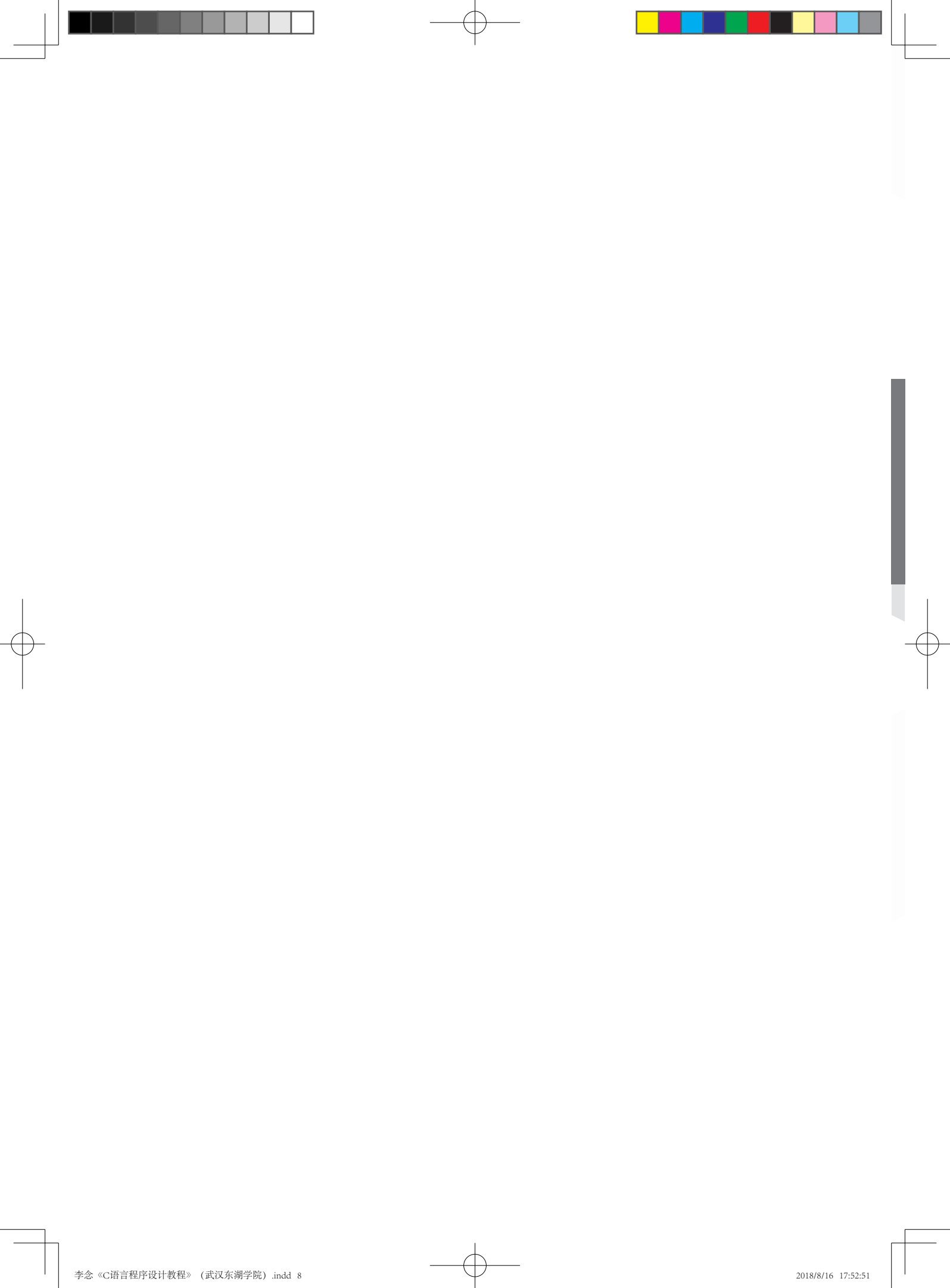
6.1 数组简介	118
6.2 一维数组	118
6.3 二维数组	123
6.4 字符数组	130
本章小结	143
自测题.....	143
实验6 数组	149

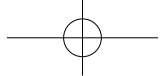
第7章 函数156

7.1 函数概述	157
7.2 函数的定义	158
7.3 函数调用	161
7.4 函数的嵌套调用和递归调用	168
7.5 用带参数的宏定义代替公式型函数	173
7.6 作用域和存储类别	175
本章小结	186
自测题.....	187
实验7 函数	196



第8章 指针	203
8.1 地址与指针	204
8.2 指针与数组	208
8.3 用字符指针操作字符串	218
8.4 指针与函数	219
本章小结	223
自测题	224
实验8 指针	233
第9章 结构体和共用体	238
9.1 定义结构体类型	239
9.2 定义和使用结构体变量	240
9.3 结构体数组	246
9.4 结构体指针	249
9.5 结构体与函数	251
9.6 共用体	254
本章小结	260
自测题	260
实验9 结构体与共用体	266
第10章 文件操作	270
10.1 C语言文件概述	271
10.2 打开和关闭文件	273
10.3 文件读写操作	276
10.4 随机读写文件	284
本章小结	286
自测题	286
实验10 文件操作	291
参考文献	296

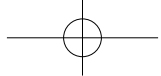




第 1 章

C 语言基础知识

C 语言是一种通用的编程语言，深受广大科技工作者喜爱。本章主要介绍 C 语言的历史发展、C 语言的特点、为什么学习 C 语言、程序设计的基本概念、C 语言程序的组成和 C 语言程序的开发过程，最后介绍 Visual C++6.0 集成环境下的上机操作过程和调试方法。学习本章的目的是使读者对 C 语言和程序设计有一个概略的了解，并掌握上机运行简单程序的操作步骤。



1.1 C语言简介

本节来了解一下 C 语言的起源以及与其他几种语言的关系，再介绍 C 语言的标准与版本和特点，最后从 C 语言的市场占有率等方面了解为什么必须学好 C 语言。

1.1.1 C 语言的发展历史及以 C 语言为基础的其他语言

1963 年，英国剑桥大学在原来的 ALGOL 语言的基础上陆续推出了 CPL 语言和简化的 BCPL 语言，1970 年美国贝尔实验室以 BCPL 语言为基础，设计出了 B 语言，并用 B 语言改写了第一个用汇编语言编写的 UNIX 操作系统软件，1972 年在 B 语言基础上，贝尔实验室 Dennis Ritchie 等经过进一步精练推出了 C 语言，并用 C 语言重新改写了 UNIX 的内核。C 语言一经推出，其强大的功能，立即引起广泛关注，C 语言的编译程序也被移植到 IBM 360/370、Honeywell 11 以及 VAX-11/780 等多种计算机上，迅速成为应用最广泛的系统程序设计语言。

为了克服 C 语言存在的缺点，1983 年贝尔实验室对 C 语言进行改进和扩充，将“类”的概念引入了 C 语言，构成了最早的 C++ 语言。后来，Stroustrup 和他的同事们又为 C++ 引进了运算符重载、引用、虚函数、多重继承、模板、异常等许多特性，使之更加精炼，并于 1989 年后推出了 AT&T C++ 2.0 版。

C# 是微软公司在 2000 年 7 月发布的一种全新且简单、安全、面向对象的程序设计语言，是专门为 .NET 的应用而开发的语言。C# 是由 C 和 C++ 衍生出来的面向对象的编程语言。它在继承 C 和 C++ 强大功能的同时去掉了一些它们的复杂特性。

Java 语言去掉了 C 语言中的指针运算、结构体等功能，减少了平常出错率的 50%，让 Java 的语言功能更加精炼。Java 实现了 C++ 的基本面向对象技术，并有一些增强。Java 与 C 和 C++ 直接相关，Java 继承了 C 的语法，Java 的对象模型是从 C++ 改编而来的。Java 编程中涉及指针的编程必须结合 C 语言才能完成。

还有其他许多语言如 PHP 和 Perl 也是以 C 语言为基础。

1.1.2 C 语言的特点

C 语言功能丰富、表达能力强、使用灵活方便、应用面广、目标程序效率高、可移植性强，既具有高级语言的优点，又具有低级语言的特点，特别适合于编写系统软件，如 UNIX，LINUX 等操作系统，也适合于应用软件的开发。C 语言还被广泛地移植到了各类型计算机上，从而形成了多种版本的 C 语言。归纳起来，C 语言具有以下主要特性。

- (1) C 语言运算符丰富，共有 34 种运算符。
- (2) C 语言简洁、紧凑，使用方便、灵活。(C89 标准中只有 32 个关键字，C99 标

准增加了6个新关键字,9种控制语句。)一般而言,学习C语言的过程就是学习掌握关键字、控制语句和运算符的过程。

(3) C语言的类型丰富。既有系统定义的简单类型,也有用户自定义的构造类型(数组类型、结构体类型、共用体类型等)。

(4) 利用三种类型简单的控制结构(顺序结构、选择结构、循环结构)就能实现任何复杂结构。采用函数作为程序的模块单位,便于实现程序的模块化。

(5) C语言能进行位(bit)操作。在软件需要对硬件进行操作的场合,用C语言明显优于其他高级语言。C语言具有强大的功能且适合多个领域的开发,特别适合开发底层(例如显卡驱动程序、打印机驱动程序等硬件驱动程序)或者系统级的应用软件。

(6) 生成目标代码质量高,速度快。一般只比汇编语言生成的目标代码效率低10%—20%。

(7) C语言指针提供了与具体机器无关的地址算术运算,用C语言编写的程序可移植性好。一般不用修改或者简单修改就能用于各种型号计算机和各种操作系统。

1.1.3 C语言的市场占有率与为什么学习C语言

1. C语言的市场占有率

在100多种编程语言中(占市场份额2%以上的不足12种,其中C、C++、Java、C#占了50%以上的总份额,而占市场0.2%左右份额的语言大约有50多种。),C语言的市场份额近10年来长期保持在15%至20%之间,以前更多。在多种编程语言中,用处各有侧重,但C语言总是最基础的应用,尤其是其指针和结构体的应用。C语言市场占有率多年来一直排在前列。

2. 学习C语言的原因

(1) 许多高级语言都是以C语言为基础的,学了C语言,就能学习许多的高级语言。

(2) 设备驱动程序和操作系统很适合用C语言来编写。编写微控制器程序,C语言更易控制数据的存储位置和初始化。

(3) ANSI标准提高了C程序在不同计算机之间的可移植性。

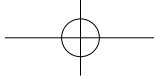
(4) C程序与其他语言编写的程序相比,实现相同的功能,代码行数更少,运行效率更高。

(5) C语言历史悠久,它有广泛的使用团体和大量的现成代码可以利用,能在已有程序的基础上,快速高效的编写新的算法和函数。

(6) 可以实现底层的计算、移位和屏蔽等操作。即使是“高层”的汇编语言,也可方便地提供汇编接口,是编写操作系统和嵌入式软件的良好选择。

(7) C语言是一个开源组织的语言,其产物就是——Linux。会C语言,就能参加该组织并向众多的开源组织投稿,如给Source Forge投稿。

(8) C语言是唯一的一个阐述指针本质的语言。而C#和Java干脆跳过了指针,但



是指针确实使 C 语言变得更加强大。

(9) 任何微处理器的设备都支持 C 语言。从微波炉到手机，都是由 C 语言技术来推动的。

随着计算机网络飞速发展，特别是 Internet 的出现，计算机通信显得尤其重要，而通信程序和游戏程序的编写首选就是 C 语言，C 语言也是编程企业招收员工必考的语言之一。

近来物联网风行世界，数据挖掘技术和数据仓库技术的发展，C 语言的云计算功能，让 C 语言重新回到了市场占有率的第一把交椅，并长盛不衰。

总之，C 语言是一门通用的、模块化的和程序化的编程语言，被广泛应用于操作系统和应用软件的开发。由于 C 语言的高效性和可移植性，适应于不同硬件和软件平台，深受程序员的青睐。

1.2 程序设计的基本概念

本节介绍程序、程序设计、算法和结构化程序设计几个概念。

1.2.1 程序和程序设计

1. 程序

程序是为实现特定目标或解决特定问题而用计算机语言编写的命令序列的集合。计算机程序是软件开发人员按用户需求、用程序设计语言开发的、为实现预期目的、适合计算机执行的指令（语句）序列的集合。程序执行的效率与数据的存储结构密切相关，计算机程序分为系统程序和应用程序两大类。程序就是为使计算机执行一个或多个操作，或执行某一任务，按序设计的指令的集合。

2. 程序设计语言

程序设计语言是编写程序的工具，是软件系统的重要组成部分，又称为计算机语言。程序设计语言可以分为机器语言、汇编语言和高级语言 3 大类。

(1) 机器语言。机器语言是由 0 和 1 组成的系列代码，用机器语言编写的程序称为机器语言程序，计算机可以直接执行。机器语言是面向机器的程序设计语言。

(2) 汇编语言。汇编语言是用助记符代替机器语言的二进制操作码，用地址符号或标号代替地址码，以方便记忆和编辑的一种低级语言。

例如：实现 1+1 要写成

```
MOV AX,1
```

```
ADD AX,1
```

使用汇编语言编写的程序，机器不能直接识别，要由一种程序将汇编语言翻译成

机器语言，这种起翻译作用的程序叫汇编程序，汇编程序是系统软件中的语言处理系统软件。汇编程序把汇编语言翻译成机器语言的过程称为汇编。机器语言和汇编语言属于低级语言，低级语言依赖于具体的机器类型，移植困难，学习较难，推广慢。

(3) 高级语言。高级语言比较多，现在可以在计算机上应用的编程语言有 100 多种，如 Basic、Java、C 语言等。高级语言有两种执行方式：解释方式和编译方式。解释方式有如口译，每输入一句便翻译执行一句，如 Basic 语言。编译方式则必须将程序编译生成目标代码（后缀 obj），再生成可执行的机器语言（后缀 exe）文件，才能执行。高级语言类似数学语言，接近自然语言，容易编写、容易理解并且不依赖于具体的计算机类型，可移植性强。

3. 程序设计

程序设计是给出解决特定问题程序的过程。程序设计往往以某种程序设计语言为工具，给出这种语言编写的程序。程序设计过程应当包括分析问题、确定数据结构、设计算法、编写程序、调试测试和编写程序文档等不同阶段。

4. 程序设计的步骤

(1) 分析问题。对于接受的任务要进行认真地分析，研究所给定的条件，分析最后应达到的目标（输出的数据格式、输出到何处、怎么样输出等）找出解决问题的规律，选择解题的方法，完成实际问题。

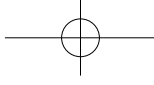
(2) 确定数据结构。根据任务的要求、指定输入数据和输出结果，确定存放数据的类型和数据的组织形式，即数据结构。

(3) 设计算法。设计出解题的方法和具体步骤，这就是设计算法。

(4) 编写程序。根据指定的数据结构和算法，用一种计算机语言编写程序，对源程序进行编辑、编译和连接。

(5) 调试测试。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。不合理要对程序进行调试，调试就是通过上机发现和排除程序中错误的过程。调试不仅要输入正确的数据、也要输入错误的数据和临界数据，不仅要达到有输出结果，还要使输出结果的数据类型、有效数据位数、输出格式完全正确，完全符合问题的要求，才能算基本完成。

(6) 编写程序文档。绝大多数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，内容应该包括：程序名称、程序功能、运行环境、程序的装入和启动、需要输入的数据及使用注意事项等。



1.2.2 算法

1. 算法的概念

对于计算机程序可处理的问题来说，程序中所用到的数据以及对这些数据的类型和数据组织形式的描述称为“数据结构”，对数据处理所用的方法和步骤等操作的描述称为“计算机算法”。数据结构可分为线性数据结构和非线性数据结构，计算机算法可分为数值运算算法和非数值运算算法。作为程序设计人员，必须认真考虑和设计数据结构和操作步骤（即算法）。著名计算机科学家沃思（Nikiklaus Wirth）提出一个公式：

程序 = 数据结构 + 算法

直到现在，对于面向过程的程序设计来说这个公式依然适用。

【实例 1-1】

计算并输出半径为 r 的圆的面积。

分析其程序实现需要包括的两方面信息：数据结构和算法。

（1）数据结构。该实例要求计算半径为 r 圆的面积。需要的数据包括半径、圆的面积以及圆周率的值，其中圆周率为浮点型常量，半径和圆的面积为未知量，这就需要准备两个变量分别存放半径和圆的面积。通过分析可以看出，这两个变量的类型为浮点型比较合适。所以该示例用到的数据结构比较简单，即 3 个单独的浮点型数据。

（2）算法。根据题目要求，用 r 表示半径， $area$ 表示面积，其算法如下：

步骤 1：输入半径 r 的值；

步骤 2：根据圆的面积公式计算 $area$ 的值；

步骤 3：输出圆的面积计算结果。

上述算法描述基本准确，但也有一些缺陷，比如当输入的半径 r 为非法的数值时（如输入的是字母），计算结果将是异常的数，所以随着内容的逐步推进，可以对上述算法进一步优化。

此外，对于同一个问题可以有不同的解题方法和步骤，也就是有不同的算法，算法有优劣，一般来说，算法的设计应充分考虑执行效率和内存开销，即算法的时间复杂度和空间复杂度。

2. 算法的描述

算法的描述有多种方法，常用的算法描述方法有自然语言、流程图、N-S 图和伪代码等，其中最为常见的是流程图和 N-S 图。

（1）流程图描述。流程图描述算法是用一些规定图框表示各种操作，用箭头表示算法流程，这种描述方法直观形象、易于理解。美国国家标准学会（ANSI）规定了一些常用的流程图符号，现已被软件开发人员普遍采用。

【实例 1-2】

计算 $1+2+3+\cdots+100$ 的和的流程图如图 1-1 所示，传统流程图使用的图例如图 1-2 所示。

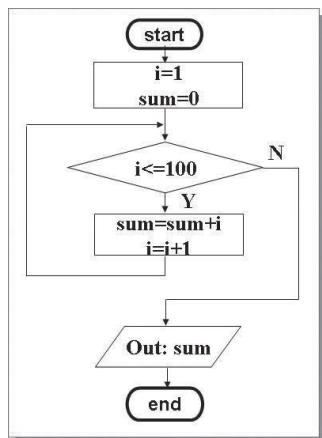
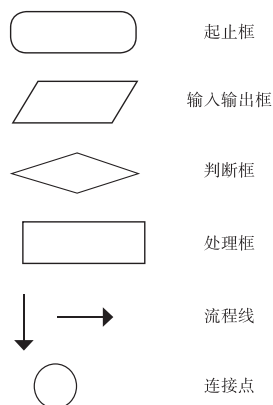
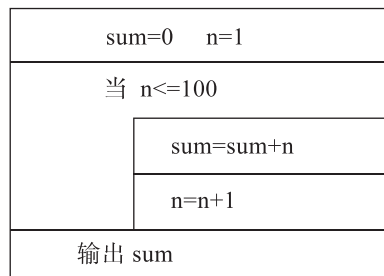
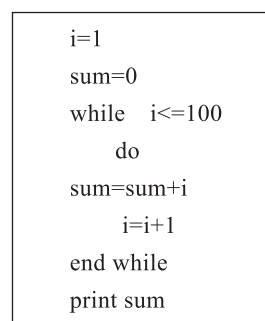
图1-1 计算 $1+2+3+\cdots+100$ 的流程图

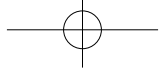
图1-2 传统流程图使用的图例

用传统的流程图表示算法的优点是直观易懂，简单方便；缺点是传统流程图对于流程线的走向没有任何限制，可以任意转向，绘制流程图时费时费力且不易修改和阅读。带箭头的线段是流程线，它是控制流程的，所以又称为控制线。

(2) N-S图描述。1973年，美国学者 I.Nassi 和 B.Shneiderman 提出了另一种流程图描述方法，并以他们的名字命名为 N-S 结构化流程图。在这种流程图中，完全去掉了带箭头的流程线，全部算法写在一个矩形框内，在该矩形框内还可以包含其他从属于它的矩形框。计算 $1+2+3+\cdots+100$ 的和的 N-S 图如图 1-3 所示，伪代码描述如图 1-4 所示。

图1-3 计算 $1+2+3+\cdots+100$ 的N-S图图1-4 计算 $1+2+3+\cdots+100$ 的伪代码描述

用 N-S 图表示算法的优点是去掉了流程线，直观易懂，容易让学习的人员养成利用结构化方法构造算法的习惯，但修改流程图仍然比较麻烦。



1.2.3 结构化程序设计

结构化程序设计 (structured programming) 是指以模块功能和处理过程设计为主的程序设计方法。它的主要观点是采用自顶向下、逐步求精及模块化的程序设计, 任何程序都可由顺序、选择和循环 3 种基本控制结构构造。

1. 结构化程序设计的原则方法

(1) 自顶向下: 程序设计时, 先考虑总体, 再考虑细节; 先考虑全局目标, 后考虑局部目标。先从最上层总目标开始设计, 不要一开始就追求细节。

(2) 逐步求精: 逐步精细化, 对复杂问题, 应设计一些子目标作为过渡, 逐步细化。

(3) 模块化设计: 把程序要解决的总目标分解为子目标, 再进一步分解为具体的小目标, 小目标又称为模块。为了使模块尽可能独立, 要求模块内聚程度要尽量高, 且模块间的耦合程度要尽量低。

(4) 结构化编码: 不得使用非结构化的跳转语句 (例如 goto 等)。

2. 结构化程序设计的 3 种基本控制结构

结构化程序设计都可由顺序、选择和循环 3 种基本控制结构。

(1) 顺序结构。顺序结构是最基本的构造结构, 当执行由这些语句构成的程序时, 按照这些语句在程序中的先后顺序执行, 没有分支, 没有转移, 没有步骤之间的相互约束。顺序结构流程图如图 1-5(a) 所示。

(2) 选择结构。选择结构就是根据不同的条件去执行不同分支语句的程序段。此结构中必包含一个判断条件, 根据判断结果从两种或多种路径中选择其中的一条执行。选择结构流程图如图 1-5(b) 所示。

(3) 循环结构。循环结构就是根据给定的条件, 判断是否要重复执行某一相同或类似代码的程序段。循环结构流程图如图 1-5(c) 所示。图 1-5(c) 左图是当型循环, 该循环的特点是, 当给定的条件成立时, 就执行循环体, 否则不执行。图 1-5(c) 右图是直到型循环, 该循环的特点是, 执行循环体, 直到给定的条件成立, 就不再执行循环。

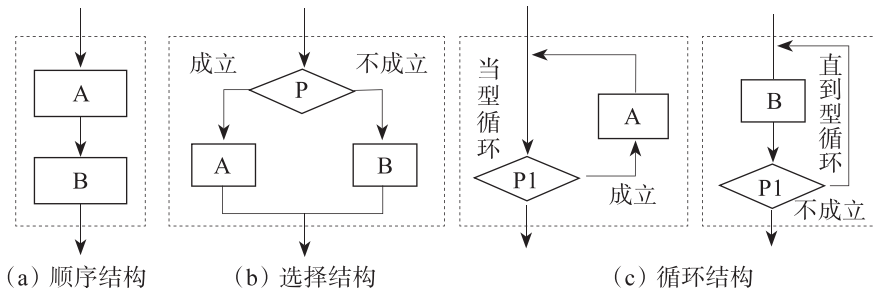


图1-5 3种基本控制结构的流程图

由以上 3 种基本结构构造的算法结构可以解决任何复杂的问题, 由它们所构成的程序称为结构化程序。

1.3 C语言程序的组成

本节通过 C 语言程序实例，对 C 程序作简单介绍并了解其构成。

1.3.1 C 程序的简单介绍

先来看几个简单的 C 程序。

【实例 1-3】

在屏幕上显示“欢迎进入 C 语言的世界!”。

设计思想：

仅由 main() 函数调用 printf() 标准库函数，输出这行文字构成 C 语言程序。

程序代码：

```
/* 显示一行文字，文件名 new1-1.c */
#include <stdio.h> /* 预处理，以 # 开头，stdio.h 是输入输出头文件 */
int main() /* 定义整型主函数 */
{
    printf(" 欢迎进入 C 语言的世界!  \n"); /* 输出一行文字 */
    return 0; /* 程序执行毕，控制权返回操作系统 */
}
```

运行结果：

```
欢迎进入 C 语言的世界!
```

该实例看上去很简单，却体现了 C 语言程序基本程序框架的两大部分：

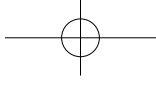
第一部分称之为“编译预处理”，如实例 1-3 程序代码中的程序段：

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

第二部分称之为函数组，如实例 1-3 程序代码中的程序段：

```
int main( )
{
    printf(" 欢迎进入 C 语言的世界!  \n");
    return 0;
}
```

编译预处理以“#”开头，其作用是为后面的函数做准备工作。实例 1-3 中的编译预处理部分只有一条头文件包含命令“#include <stdio.h>”，其含义是在程序中包含标准输入输出头文件 stdio.h，该头文件中声明了输入和输出库函数及其他信息，这意味着在后面的程序中将用到该文件中的内容。



C 语言程序设计教程

函数组由多个函数构成，函数是构成 C 语言程序的基本单位，多个函数共同协作完成程序要实现的功能。函数组中必须包括一个 “main()” 主函数，且 C 语言程序中有且仅有一个主函数，整个程序的执行从主函数开始，以主函数为核心展开。此外，函数组中除了主函数外还包括库函数和用户自定义的函数。

实例 1-3 中的函数组部分只有一个主函数 “main()”，主函数调用库函数 “printf()” 在屏幕上输出 “欢迎进入 C 语言的世界!” 提示信息，库函数使用之前必须在编译预处理部分包含其对应的头文件，所以在示例程序的开始就出现了 “#include <stdio.h>”。

除了主体框架的编译预处理和函数组以外，在程序中还允许为程序添加注释，以增强程序的可读性。实例 1-3 中以 “/* */” 标识的就是程序中的注释。

【实例 1-4】

从键盘上输入矩形的长和宽，计算并在屏幕上输出该矩形的周长。

设计思想：

编译预处理部分为一条头文件包含命令 “#include <stdio.h>”，函数组部分包括一个主函数 main()、一个输出函数 printf() 和一个输入函数 scanf()，主函数通过数据准备、数据输入、数据计算、数据结果等语句实现了题目的要求。

程序代码：

```
/* 计算矩形周长，文件名 new1-2.c */
#include <stdio.h>          /* 包含标准输入输出头文件 */
int main()                  /* 主函数 */
{
    int a,b,c;              /* 定义整型变量，长 a 宽 b 周长 c */
    printf(" 请输入矩形的长和宽: "); /* 调用输出函数，显示提示语 */
    scanf("%d%d",&a,&b);    /* 数据输入，从键盘输入 a、b 值 */
    c=2*(a+b);              /* 数据计算，将计算的周长赋值给 c */
    printf(" 该矩形周长为: %d\n",c); /* 调用输出函数输出结果 */
    return 0;
}
```

运行结果：

```
请输入矩形的长和宽: 5 10
该矩形周长为: 30
```

通过上面两个实例的描述和解读，相信读者对 C 语言程序宏观框架有了一个大致了解，下面总结如下：

- (1) C 语言程序基本框架包括编译预处理和函数组两部分。
- (2) 编译预处理是程序编译之前的准备工作，以 “#” 开头。

(3) 函数组包括主函数、库函数和用户自定义函数。函数是构成 C 语言程序的基本单位, `main()` 称为主函数, C 语言程序中有且仅有一个主函数, 整个程序的执行从主函数开始, 以主函数为核心展开; C 语言标准函数库中提供了大量功能丰富的库函数, 当使用到相应的库函数时, 需要在编译预处理中包含相应的头文件, 用户也可以根据需要编写具有特定功能的函数, 称为用户自定义函数。

(4) 在 C 语言中的适当位置可添加注释, 以增强程序的可读性。在 Visual C++6.0 编程环境中, 可用 “//” 作为程序单行注释的起始符号, 也可以使用 “/*” 和 “*/” 作为单行或多行注释的起始和终止符号。

1.3.2 C 程序的构成

用 C 语言编写的程序称为 C 程序, 在程序宏观框架结构下, C 语言程序由标识符、函数和语句等表述形式构成, 最终形成完整的 C 语言程序代码。

1. 标识符

在实例 1-4 的程序代码中, 可以看到由 `void`、`main`、`int`、`printf`、`scanf`、`a`、`b` 和 `c` 等一系列符号构成的语句和函数, 这些符号统称为标识符。标识符是用来标识程序中某个对象名字的字符序列, 这些对象可以是语句、数据类型、函数、变量和常量等。C 语言把标识符分为 3 类: 关键字、预定义标识符和用户自定义标识符。

(1) 关键字。在 C 语言中, 为了定义变量、表达语句功能或说明某些信息的存储类型, 必须用到一些具有特殊意义的标识符, 这些标识符就是关键字。如实例 1-4 程序代码中的 `void` 和 `int`。

C 语言中关键字主要有以下两类:

①类型说明符: 用来说明变量、函数的类型, 如 `int`、`float`、`char` 和 `void` 等。

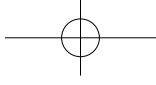
②语句定义符: 用来表示一个语句的功能, 如 `if`、`for` 和 `while` 等。

(2) 预定义标识符。预定义标识符是指已经被 C 语言系统预先定义好了的具有特定含义的标识符, 如实例 1-4 程序代码中的函数名 `printf` 和 `scanf`, 以及预处理命令 `define` 和 `include` 等。

(3) 用户自定义标识符。在编写程序过程中, 用户给自定义的符号常量、变量、函数、数组和类型等起的名字就是用户自定义标识符。C 语言规定, 所有的用户标识符必须先定义或说明后再使用。

用户自定义标识符的命名有其规则: C 语言规定, 用户标识符由字母 (A-Z, a-z)、数字 (0-9) 和下划线 “_” 组成, 并且首字符不能是数字, 通常以下划线开头的标识符是编译系统专用的, 所以在编写 C 语言程序时, 最好不要使用以下划线开头的标识符。

在用户自定义标识符时还应注意以下几点:



① C 语言对大小写字符敏感，所以在编写程序时要注意大小写字符的区分。例如：对于 `max` 和 `Max` 这两个标识符来说，C 语言会认为这是两个完全不同的标识符。

② 不能把 C 语言关键字作为用户自定义标识符。

③ 通常不使用预定义标识符作为用户自定义标识符，因为这样会失去系统规定的原意，造成二义性。

④ 用户自定义标识符的命名应做到简洁明了，尽量做到“见名知意”，这样便于程序的阅读和维护。例如用 `length` 表示长度，用 `sum` 表示求和。

2. 函数和语句

函数是构成 C 语言程序的基本单位，而语句则是函数的重要组成部分。函数结构形式如下所示：

```
函数返回值类型函数名（形式参数）// 函数首部
{
    说明语句；           // 函数体
    可执行语句；
}
```

函数的第一行称为函数首部，大括号“{}”括起来的部分称为函数体，函数体由若干语句组成，函数体使用一对大括号“{”和“}”作为定界符。

根据语句在程序中所起的作用可分为说明语句和可执行语句，通过这些语句实现对数据的描述和操作。通常每行只写一条语句，每个语句以分号“;”结束，如果没有写“;”，在程序编译时会出现“missing ';'”(忘记';')的错误提示。

1.4 程序的开发过程及环境

1.4.1 程序开发过程

用 C 语言编写的程序称为源程序，它不能被计算机直接识别和执行，为此，还需要一种担任翻译工作的程序，即编译程序，它可以把 C 语言程序代码转换为计算机能够识别和执行的二进制目标代码。

开发一个 C 程序，一般要经过编辑、编译、连接组建和运行四个步骤。

1. 编辑源程序

编辑源程序是指在文本编辑工具软件中输入和修改 C 语言源程序，最后以文本文件的形式存放在磁盘上。编辑生成的文件以文本形式存储，文件后缀名为 `.c`，也称为

C 源程序文件。注意在新建文件时,就必须输入文件全名和后缀名,(如果不输入后缀名.c,在 VC++6.0 中会自动以后缀名 .cpp 保存文件,执行 C++ 的语法检查程序。)C 源程序文件以 ASCII 码形式存储,计算机不能直接执行。

2. 编译源程序,生成目标程序

编译是将已编辑好的源程序翻译成二进制目标程序。编译是由系统本身的编译程序来完成的,编译过程将对源程序进行语法检查,当发现错误时,会提示错误的类型和出错的位置,以使用户修改。直至未发现语法错误时,会自动形成扩展名为“.obj”的目标程序。

3. 连接目标程序及其相关模块,生成可执行程序

一个 C 语言应用程序,可能包含有 C 语言标准库函数和许多模块,而各个模块往往是单独编译的,因此,经编译后得到的目标程序还不能直接执行,需要把编译好的各个模块的目标程序与系统提供的标准库函数进行连接,生成扩展名为“.exe”的可执行程序。

连接过程由系统提供的连接程序完成,如果连接过程中出现错误信息,则需要修改错误后重新进行编译和连接,直至生成可执行程序。

4. 运行可执行程序

运行程序,并检查运行结果。如果是算法错误,只能回到第一步修改源程序,再重新编译、连接和运行,直到得到正确结果。

总之,C 语言程序开发主要经过编辑、编译、连接和执行四个步骤,其完整过程描述如图 1-6 所示。

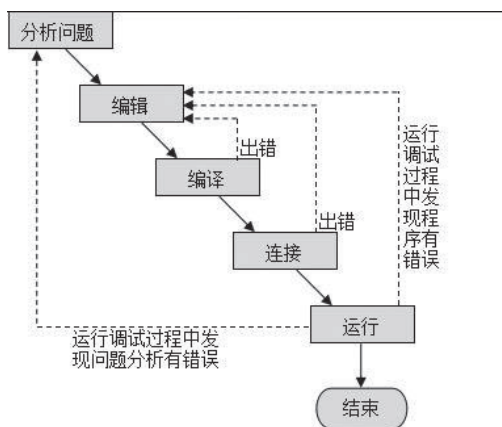


图1-6 C语言程序开发过程

1.4.2 Visual C++ 6.0 集成开发环境

C 语言的集成开发环境比较多,现在 windows 环境下常用 Visual C++ 6.0。本教材

主要介绍英文版的 Visual C++ 6.0，其他集成开发软件不再一一介绍。

1. 启动 VC++6.0

选择“开始”→“程序”→“Microsoft Visual Studio 6.0”→“Microsoft Visual C++ 6.0”启动运行 VC++6.0，进入到它的集成开发环境窗口。如图 1-7 所示。

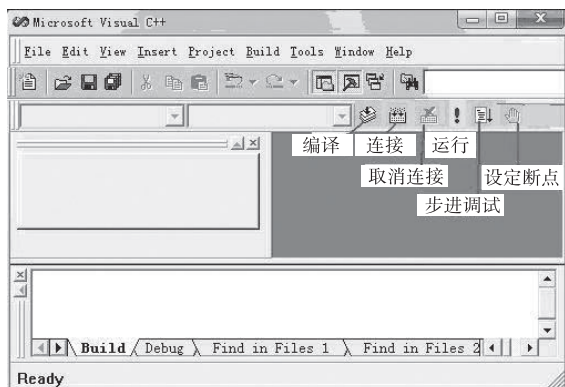


图1-7 Microsoft Visual C++ 6.0开发环境窗口

2. 编辑——C 源程序

在 VC++6.0 窗口，选择菜单“File | New”，弹出如图 1-8 所示的新建文件对话框。

在弹出的对话框中，单击上方的选项卡“File”，选择“C++ Source File”，在“File name”一栏中填写文件名（例如 hello.c），在“Location”一栏中选择或填写文件存放的位置（目录）。然后单击“OK”按钮。

一定要指定文件保存的目录，不要用系统缺省目录或者随便放在根目录或其他目录下。然后在打开的编辑区中输入实例 1-3 中的程序代码。

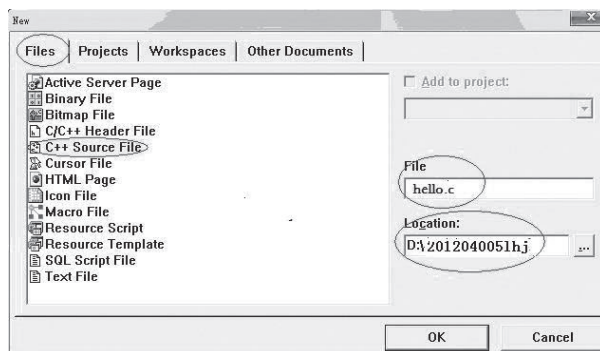


图1-8 新建文件对话框

3. 编译——产生目标代码 .obj 文件

选择菜单“Build | Compile”或单击工具栏上的编译图标，如果编译没有错误和警告就会显示“hello.obj -- 0 error(s), 0 warning(s)”。这表示目标代码文件建成，可以在

文件保存文件夹的下级文件夹 debug 中看到 hello.obj。

4. 连接(组建)——生成可执行 .exe 文件

选择菜单“Build | Build”，这时 VC++6.0 将弹出一个对话框，说明这个命令需要建一个工程(Project)，问：“This build command requires an active project workspace. Would you like to create a default project workspace?”，是否创建一个默认的工程？一般单击“Yes”。这时在左边灰色区域建立白色工程区，显示工程名。

5. 执行——编辑器下面的窗口开始显示编译的结果

如果看到信息“hello.exe -- 0 error(s), 0 warning(s)”，表示编译没有错误和警告，编译已经通过，单击快捷工具栏上的红色的感叹号“!”(运行)按钮(或者选择菜单“Build | Execute”或按 Ctrl-F5)，VC++6.0 将自动打开一个显示结果的黑色窗口，显示“欢迎进入 C 语言的世界!”。这表示运行正常。

本章小结

C 语言是一种通用的程序设计语言，它具有很多突出的特性，利用 C 语言能够编写出功能强大、代码简洁、结构规范，可移植性强的程序，深受广大程序员的喜爱。

本章首先介绍了 C 语言的发展历史、C 语言的特点，然后从程序设计的基本概念、C 语言程序结构的组成、C 程序的开发过程等方面对 C 语言进行了介绍，最后对 Visual C++6.0 集成环境作了简单介绍。学习本章的目的是使读者对 C 语言和程序设计有一个概略的了解，并掌握上机编译运行简单程序的操作步骤。

自测题

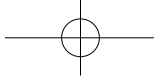
一、单项选择题

1. 为了避免传统流程图在描述程序逻辑时的灵活性，提出了用方框图来代替传统的流程图，通常也把这种流程图称为_____。

- A. PAD 图
- B. N-S 图
- C. 结构图
- D. 数据流图

2. 结构化程序设计主要强调的是_____。

- A. 程序的规模
- B. 程序的效率
- C. 程序设计语言的先进性
- D. 程序的易读性

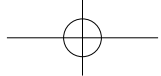


3. C 语言中可处理的文件类型是 ____。
 - A. 文本文件和数据文件
 - B. 文本文件和二进制文件
 - C. 数据文件和二进制文件
 - D. 数据代码文件
4. C 语言程序经过链接以后生成的可执行文件的扩展名是 ____。
 - A. .c
 - B. .o
 - C. .exe
 - D. .obj
5. C 语言程序由 ____ 组成。
 - A. 函数
 - B. 主程序和子程序
 - C. 子程序
 - D. 过程
6. 以下叙述中正确的是 ____。
 - A. C 语言的源程序不必通过编译就可以直接运行
 - B. C 语言中的每条可执行语句最终都将被转换成二进制的机器指令
 - C. C 源程序经编译形成的二进制代码可以直接运行
 - D. C 语言中的函数不可以单独进行编译
7. C 语言具有低级语言的能力, 主要指的是 ____。
 - A. 程序的可移植性
 - B. 具有控制流语句
 - C. 能直接访问物理地址, 可进行位操作
 - D. 具有现代化语言的各种数据结构
8. 下面哪一项不属于 C 语言集成开发环境包含的程序 ____。
 - A. 编辑程序
 - B. 编译程序
 - C. 汇编程序
 - D. 调试程序
9. 程序流程图中带有箭头的线段表示的是 ____。
 - A. 图元关系
 - B. 数据流
 - C. 控制流
 - D. 调用关系
10. 下面所述正确的是 ____。
 - A. 目标文件可以直接执行
 - B. C# 是 C 语言的超集, 包含了 C 语言的所有内容
 - C. 编译器接收 C 语言源文件, 生成可执行程序
 - D. 可以用记事本来编写 C 语言源程序
11. 下面不属于 C 语言特点的是 ____。
 - A. C 语言是一种面向对象程序设计语言
 - B. C 语言程序执行效率高
 - C. C 语言具有自我扩展能力
 - D. C 语言具有可移植性
12. 所有 C 函数的结构都包括的三部分是 ____。
 - A. 语句、花括号和函数体
 - B. 函数名、语句和函数体
 - C. 函数名、形式参数和函数体
 - D. 形式参数、语句和函数体 \

13. 以下叙述中错误的是_____。
- A. C语言的可执行程序是由一系列机器指令构成的
 - B. 用C语言编写的源程序不能直接在计算机上运行
 - C. 通过编译得到的二进制目标程序需要连接才可以运行
 - D. 在没有安装C语言集成开发环境的机器上不能运行C源程序生成的.exe文件
14. 以下选项中关于程序模块化的叙述错误的是_____。
- A. 可采用自底向上、逐步细化的设计方法把若干独立模块组装成所要求的程序
 - B. 把程序分成若干相对独立, 功能单一的模块, 可便于重复使用这些模块
 - C. 把程序分成若干相对独立的模块可便于编码和调试
 - D. 可采用自顶向下、逐步细化的设计方法把若干独立模块组装成所要求的程序
15. 以下叙述正确的是_____。
- A. 在对一个C程序进行编译的过程中, 可发现注释中的拼写错误
 - B. 在C程序中, main()函数必须位于程序的最前面
 - C. C语言本身没有输入输出语句
 - D. C程序的每行中只能写一条语句
16. 最早开发C语言是为了编写下面哪一种操作系统_____。
- A. DOS
 - B. Windows95
 - C. UNIX
 - D. Linux
17. C语言可执行程序的开始执行点是_____。
- A. 程序中第一条可执行语句
 - B. 程序中第一个函数
 - C. 程序中的main()函数
 - D. 包含文件中的第一个函数
18. 计算机能直接执行的程序是_____。
- A. 源程序
 - B. 可执行程序
 - C. 目标程序
 - D. 汇编程序
19. 结构化程序设计的基本原则不包括_____。
- A. 自顶向下
 - B. 逐步求精
 - C. 模块化
 - D. 多态性
20. 一个完整的C源程序是_____。
- A. 由一个主函数或一个以上的非主函数构成
 - B. 由一个且仅由一个主函数和零个以上(含零)的非主函数构成
 - C. 要由一个及一个以上主函数和非主函数构成
 - D. 由一个且只由一个主函数或多个非主函数构成

二、填空题

1. C语言链接器把编译生成的目标代码、启动代码和代码连接起来生成_____。
2. 高级语言程序需要被编译器或编译程序翻译成_____。



3. _____是 C 语言程序的基本模块。
4. C 语言预处理命令都是以_____开头。
5. 结构化程序设计的原则是自顶向下、_____、模块化设计和结构化编码。
6. stdio.h 头文件包含了_____标准的声明。
7. 符合结构化原则的三类基本控制结构是：选择结构、循环结构和_____。

三、判断题（正确的打“T”，错误的打“F”）

- () 1. 符合良好程序设计风格的要求是程序效率第一，清晰第二。
- () 2. C 语言程序执行效率高。
- () 3. C 语言只能用来编写操作系统。
- () 4. 在 C 语言中，程序的模块化是利用函数实现的。
- () 5. 一条 #include 命令只能指定包含一个头文件。
- () 6. 连接器只是把编译生成的目标代码连接起来生成可执行代码。
- () 7. C 程序的语句不一定要以“;”结束。
- () 8. 一个 C 语言程序有且仅有一个 main() 主函数。
- () 9. C 程序在运行过程中所有数据都以十进制方式表示。
- () 10. 一个 C 语言程序只要编译和连接没有错误，就肯定能正确运行。

实验1 熟悉Visual C++ 6.0集成开发环境

[实验案例：编写一个程序，输出下面字符]

```
*****  
***** 欢迎学习 C 语言 *****  
*****
```

实验指导：

- (1) 启动 Visual C++ 6.0，新建一个 C 语言文件；
- (2) 输入以下程序：

```
/* 文件名 syl-1.c */  
#include <stdio.h>  
int main()  
{  
    printf( "*****\n" );  
    printf( "***** 欢迎学习 C 语言 *****\n" );  
    printf( "*****" );  
}
```