

offcn 中公教育

计算机二级·无纸化考试
C 语言·全真模拟 3 合 1

考试大纲与考点精要

中公教育优就业研究院◎编著

电子科技大学出版社

目录

第一部分 考试大纲	1
1.1 基本要求	1
1.2 考试内容	1
1.3 考试方式	2
第二部分 考点精要	3
2.1 公共基础知识	3
2.1.1 数据结构与算法	3
2.1.2 程序设计基础	5
2.1.3 软件工程基础	6
3.1 C语言概述	8
3.1.1 C语言的特点	8
3.1.2 C语言的书写格式	8
3.1.3 C语言程序文件	8
3.1.4 C语言的函数	8
3.1.5 C语言的基本词法	8
3.1.6 存储形式	9
3.1.7 注释	9
3.1.8 算法	9
3.1.9 算法的特性	9
3.1.10 头文件	9
3.2 数据类型、运算符与表达式	9
3.2.1 标识符的规则	9
3.2.2 C语言标识符的分类	10
3.2.3 常量与变量	10
3.2.4 整型数据	10
3.2.5 实型数据	10
3.2.6 字符型数据	10
3.2.7 运算符	10

3.2.8 不同类型数据间的转换与运算	11
3.2.9 自增自减运算	11
3.2.10 逗号运算	11
3.3 基本语句	11
3.3.1 C 语言语句	11
3.3.2 数据输入与输出	12
3.3.3 printf 函数	12
3.3.4 printf 函数中格式说明	12
3.3.5 scanf 函数	12
3.3.6 getchar 函数	12
3.3.7 putchar 函数	13
3.3.8 gets 函数	13
3.3.9 puts 函数	13
3.4 选择结构程序设计	13
3.4.1 if 语句	13
3.4.2 关系运算	13
3.4.3 逻辑运算	13
3.4.4 switch 语句	13
3.4.5 选择结构的嵌套	14
3.4.6 goto 语句	14
3.5 循环结构程序设计	14
3.5.1 for 循环结构	14
3.5.2 while 循环结构	14
3.5.3 do-while 循环结构	14
3.5.4 break 语句	14
3.5.5 continue 语句	15
3.5.6 循环的嵌套	15
3.6 数组的定义和引用	15
3.6.1 数组的定义	15
3.6.2 数组的初始化	15
3.6.3 元素的引用	15
3.6.4 二维数组	16
3.6.5 二维数组的初始化	16

3.6.6	二维数组的引用	16
3.6.7	数组名	16
3.6.8	字符串常量及表示	16
3.6.9	字符数组	16
3.6.10	字符串赋值	16
3.6.11	字符数组的初始化	16
3.6.12	字符串的输入与输出	17
3.6.13	字符串函数	17
3.7	函数	18
3.7.1	函数的定义	18
3.7.2	函数的返回值	18
3.7.3	函数的声明	18
3.7.4	函数的形参与实参	18
3.7.5	函数的调用	18
3.7.6	函数的嵌套调用	19
3.7.7	函数的递归调用	19
3.7.8	用户函数和标准库函数	19
3.7.9	变量的存储类别	19
3.7.10	变量的作用域和生存期	20
3.8	编译预处理	20
3.8.1	编译预处理	20
3.8.2	宏定义(不带参数的宏定义,带参数的宏定义)	20
3.8.3	预编译	21
3.8.4	“文件包含”处理	21
3.8.5	条件编译	21
3.8.6	动态存储分配	22
3.9	指针	22
3.9.1	指针与指针变量	22
3.9.2	指针变量的定义及初始化	22
3.9.3	指针变量的引用	23
3.9.4	指针的运算	23
3.9.5	指针的级别	23
3.9.6	函数的指针和指向函数的指针变量	23

3.9.7 结构体的指针与指向结构体的指针变量	23
3.9.8 用指针做函数参数	23
3.9.9 返回指针值的指针函数	24
3.9.10 指针数组、指向指针的指针	24
3.10 结构体与共同体	24
3.10.1 结构体类型及变量的定义	24
3.10.2 结构体成员的引用	24
3.10.3 结构体成员的操作	24
3.10.4 共用体类型的定义	24
3.10.5 类型重定义 typedef	25
3.11 位运算	25
3.11.1 位运算的概念	25
3.11.2 位运算符	25
3.11.3 位段	25
3.12 文件操作	26
3.12.1 文件类型指针	26
3.12.2 文本文件与二进制文件	26
3.12.3 打开、关闭文件	26
3.12.4 文件函数	26

第一部分

考试大纲

► 1.1 基本要求

- (1)熟悉 Visual C++ 6.0 集成开发环境。
- (2)掌握结构化程序设计的方法,具有良好的程序设计风格。
- (3)掌握程序设计中简单的数据结构和算法并能阅读简单的程序。
- (4)在 Visual C++ 6.0 集成环境下,能够编写简单的 C 程序,并具有基本的纠错和测试程序的能力。

► 1.2 考试内容

一、C 语言程序的结构

- (1)程序的构成,main 函数和其他函数。
- (2)头文件,数据说明,函数的开始和结束标志以及程序中的注释。
- (3)源程序的书写格式。
- (4)C 语言的风格。

二、数据类型及其运算

- (1)C 的数据类型(基本类型,构造类型,指针类型,无值类型)及其定义方法。
- (2)C 运算符的种类,运算优先级和结合性。
- (3)不同类型数据间的转换与运算。
- (4)C 表达式类型(赋值表达式,算术表达式,关系表达式,逻辑表达式,条件表达式,逗号表达式)和求值规则。

三、基本语句

- (1)表达式语句,空语句,复合语句。
- (2)输入输出函数的调用,正确输入数据并正确设计输出格式。

四、选择结构程序设计

- (1)用 if 语句实现选择结构。
- (2)用 switch 语句实现多分支选择结构。
- (3)选择结构的嵌套。

五、循环结构程序设计

- (1)for 循环结构。
- (2)while 和 do-while 循环结构。
- (3)continue 语句和 break 语句。
- (4)循环的嵌套。

六、数组的定义和引用

- (1)一维数组和二维数组的定义、初始化和数组元素的引用。
- (2)字符串与字符数组。

七、函数

- (1)库函数的正确调用。
- (2)函数的定义方法。
- (3)函数的类型和返回值。
- (4)形式参数与实际参数,参数值的传递。

- (5)函数的正确调用,嵌套调用,递归调用。
- (6)局部变量和全局变量。
- (7)变量的存储类别(自动,静态,寄存器,外部),变量的作用域和生存期。

八、编译预处理

- (1)宏定义和调用(不带参数的宏,带参数的宏)。
- (2)“文件包含”处理。

九、指针

- (1)地址与指针变量的概念,地址运算符与间址运算符。
- (2)一维、二维数组和字符串的地址以及指向变量、数组、字符串、函数、结构体的指针变量的定义。通过指针引用以上各类型数据。

- (3)用指针作函数参数。
- (4)返回地址值的函数。
- (5)指针数组,指向指针的指针。

十、结构体(即“结构”)与共同体(即“联合冶”)

- (1)用 typedef 说明一个新类型。
- (2)结构体和共用体类型数据的定义和成员的引用。
- (3)通过结构体构成链表,单向链表的建立,结点数据的输出、删除与插入。

十一、位运算

- (1)位运算符的含义和使用。
- (2)简单的位运算。

十二、文件操作

只要求缓冲文件系统(即高级磁盘 I/O 系统),对非标准缓冲文件系统(即低级磁盘 I/O 系统)不要求。

- (1)文件类型指针(FILE 类型指针)。
- (2)文件的打开与关闭(fopen,fclose)。
- (3)文件的读写(fputc,fgetc,fputs,fgets,fread,fwrite,fprintf,scanf 函数的应用),文件的定位(rewind,fseek 函数的应用)。

► 1.3 考试方式

上机考试,考试时长 120 分钟,满分 100 分。

(1)题型及分值

- ①单项选择题 40 分(含公共基础知识部分 10 分)
- ②操作题 60 分(包括程序填空题、程序修改题及程序设计题)。

(2)考试环境

- ①操作系统:中文版 Windows7。
- ②开发环境:Microsoft Visual C++6.0。

第二部分

考点精要

► 2.1 公共基础知识

2.1.1 数据结构与算法

(1) 算法的相关观念

算法是指一系列解决问题的清晰指令;4个基本特征:可行性、确定性、有穷性、拥有足够的情报;两种基本要素:对数据对象的运算和操作、算法的控制结构(运算和操作时间的顺序);设计的基本方法有列举法、归纳法、递推法、递归法、减半递推技术和回溯法。算法的复杂度——算法的时间复杂度(执行算法所需要的计算T作量);算法的空间复杂度(执行算法所需的内存空间)。

(2) 数据结构的基本概念

数据结构指相互有关联的数据元素的集合,即数据的组织形式。其中逻辑结构反映数据元素之间逻辑关系;存储结构为数据的逻辑结构在计算机存储空间中的存放形式,有顺序存储、链式存储、索引存储和散列存储4种方式。

数据结构按各元素之间前后件关系的复杂度可划分为:a)线性结构有且只有一个根结点,且每个结点最多有一个直接前驱和一个直接后继的非空数据结构;b)非线性结构是不满足线性结构的数据结构。

(3) 线性表及其顺序存储结构

线性表的基本概念:线性结构又称线性表,线性表是最简单也是最常用的一种数据结构。线性表的顺序存储结构:a)元素所占的存储空间必须连续;b)元素在存储空间的位置是按逻辑顺序存放的。

线性表的插入运算——在第*i*个元素之前插入一个新元素的步骤:a)把原来第*n*个结点至第*i*个结点依次往后移一个元素位置;b)把新结点放在第*i*个位置上;c)修正线性表的结点个数。在最坏情况下,即插入元素在第一个位置,线性表中所有元素均需要移动。

线性表的删除运算——删除第*i*个位置的元素的步骤:a)把第*i*个元素之后不包括第*i*个元素的*n-i*个元素依次前移一个位置;b)修正线性表的结点个数。

(4) 栈和队列

栈的基本概念:栈是一种特殊的线性表,其插入运算与删除运算都只在线性表的一端进行,也被称为“先进后出”表或“后进先出”表;栈顶为允许插入与删除的一端;栈底为栈顶的另一端;空栈是指栈中没有元素的栈。

栈的特点:栈顶元素是最后被插入和最早被删除的元素;栈底元素是最早被插入和最后被删除的元素;栈有记忆作用;在顺序存储结构下,栈的插入和删除运算不需移动表中其他数据元素;栈顶指针top动态反映了栈中元素的变化情况。

顺序存储和运算:入栈运算、退栈运算和读栈顶运算。

队列的基本概念:队列是指允许在一端进行插入,在另一端进行删除的线性表,又称“先进先出”的线性表;队尾是指允许插入的一端,用尾指针指向队尾元素;排头是指允许删除的一端,用头指针指向头元素的前一位置。

循环队列及其运算:所谓循环队列,就是将队列存储空间的最后一个位置绕到第一个位置,形成逻辑上的环状空间,包括入队运算与退队运算。入队运算是指在循环队列的队尾加入一个新元素。当循环队列非空($s=1$)且队尾指针等于队头指针时,说明循环队列已满,不能进行入队运算,这种情况称为“上溢”。退队运算是指在循环队列的队头位置退出一个元素并赋给指定的变量。首先将队头指针进一,然后将排头指针指向的元素赋给指定的变量。当循环队列为空($s=0$)时,不能进行退队运算,这种情况称为“下溢”。

(5) 线性链表:在定义的链表中,若只含有一个指针域来存放下一个元素地址,称这样的链表为单链表或线性链表。在链式存储方式中,要求每个结点由两部分组成:一部分用于存放数据元素值,称为数据域;另一部分

用于存放指针,称为指针域。其中指针用于指向该结点的前一个或后一个结点(即前件或后件)。

(6) 树和二叉树

树的基本概念:树是简单的非线性结构,树中有且仅有一个没有前驱的结点称“根”,其余结点分成 m 个互不相交的有限集合 $\{T_1, T_2, \dots, T_m\}$, 每个集合又是一棵树,称 $\{T_1, T_2, \dots, T_m\}$ 为根结点的子树。

父结点:每一个结点只有一个前件,无前件的结点只有一个,称为树的根结点(简称树的根);子结点:每一个结点可以有多个后件,无后件的结点称为叶子结点;树的度:所有结点最大的度;树的深度:树的最大层次。

二叉树的定义:二叉树是一种非线性结构,是有限的结点集合,该集合为空(空二叉树)或由一个根结点及两棵互不相交的左右二叉子树组成。可分为满二叉树和完全二叉树,其中满二叉树一定是完全二叉树,但完全二叉树不一定是满二叉树。

二叉树具有如下两个特点:二叉树可为空,空的二叉树无结点,非空二叉树有且只有一个根结点;每个结点最多可有两棵子树,称为左子树和右子树。

二叉树的基本性质:a.在二叉树的第 k 层上至多有 2^{k-1} 个结点($k \geq 1$);b.深度为 m 的二叉树至多有 $2^m - 1$ 个结点;c.对任何一棵二叉树,度为 0 的结点(即叶子结点)总是比度为 2 的结点多一个;d.具有 n 个结点的完全二叉树的深度至少为 $\lceil \log_2 n \rceil + 1$,其中 $\lceil \log_2 n \rceil$ 表示 $\log_2 n$ 的整数部分。

满二叉树:满二叉树是指这样的一种二叉树:除最后一层外,每一层上的所有结点都有两个子结点。满二叉树在其第 i 层上有 2^{i-1} 个结点。从上面满二叉树定义可知,二叉树的每一层上的结点数都必须都达到最大,否则就不是满二叉树。深度为 l 的满二叉树有 $2^l - 1$ 个结点。

完全二叉树:完全二叉树是指这样的二叉树:除最后一层外,每一层上的结点数均达到最大值;在最后一层上只缺少右边的若干结点。如果一棵具有 n 个结点的深度为 k 的二叉树,它的每一个结点都与深度为 k 的满二叉树中编号为 $1 \sim n$ 的结点一一对应。

二叉树的存储结构:二叉树通常采用链式存储结构,存储结点由数据域和指针域(左指针域和右指针域)组成。二叉树的链式存储结构也称二叉链表,对满二叉树和完全二叉树可按层次进行顺序存储。

二叉树的遍历:二叉树的遍历是指不重复地访问二叉树中所有结点,主要指非空二叉树,对于空二叉树则结束返回。二叉树的遍历包括前序遍历、中序遍历和后序遍历。

a)前序遍历是指在访问根结点、遍历左子树与遍历右子树这三者中,首先访问根结点,然后遍历左子树,最后遍历右子树;并且,在遍历左右子树时,仍然先访问根结点,然后遍历左子树,最后遍历右子树。前序遍历描述为:若二叉树为空,则执行空操作;否则,①访问根结点;②前序遍历左子树;③前序遍历右子树。

b)中序遍历是指在访问根结点、遍历左子树与遍历右子树这三者中,首先遍历左子树,然后访问根结点,最后遍历右子树;并且,在遍历左、右子树时,仍然先遍历左子树,然后访问根结点,最后遍历右子树。中序遍历描述为:若二叉树为空,则执行空操作;否则,①中序遍历左子树;②访问根结点;③中序遍历右子树。

c)后序遍历是指在访问根结点、遍历左子树与遍历右子树这三者中,首先遍历左子树,然后遍历右子树,最后访问根结点,并且在遍历左、右子树时,仍然先遍历左子树,然后遍历右子树,最后访问根结点。后序遍历描述为:若二叉树为空,则执行空操作;否则,①后序遍历左子树;②后序遍历右子树;③访问根结点。

(7) 查找技术

顺序查找:在线性表中查找指定的元素。最坏情况下,最后一个元素才是要找的元素,则需要与线性表中所有元素比较,比较次数为 n 。

二分查找:二分查找也称折半查找,它是一种高效率的查找方法。但二分查找有条件限制,它要求表必须用顺序存储结构,且表中元素必须按关键字有序(升序或降序均可)排列。对长度力 n 的有序线性表,在最坏情况下,二分查找法只需比较 $\log_2 n$ 次。

(8) 排序技术

a) 交换类排序法

冒泡排序:通过对待排序序列从后向前或从前向后,依次比较相邻元素的排序码,若发现逆序则交换,使较大的元素逐渐从前部移向后部或较小的元素逐渐从后部移向前部,直到所有元素有序为止。在最坏情况下,对

长度为 n 的线性表排序,冒泡排序需要比较的次数为 $n(n-1)/2$ 。

快速排序是迄今为止所有内排序算法中速度最快的一种。它的基本思想是任取待排序序列中的某个元素作为基准(一般取第一个元素),通过一趟排序,将待排元素分为左右两个子序列,左子序列元素的排序码均小于或等于基准元素的排序码,右子序列的排序码则大于基准元素的排序码,然后分别对两个子序列继续进行排序,直至整个序列有序。最坏情况下,即每次划分,只得到一个序列,时间效率为 $O(n^2)$ 。

b)插入类排序法

简单插入排序法:把 n 个待排序的元素看成为一个有序表和一个无序表,开始时有序表中只包含一个元素,无序表中包含有 $n-1$ 个元素,排序过程中每次从无序表中取 m 第一个元素,把它的排序码依次与有序表元素的排序码进行比较,将它插入到有序表中的适当位置,使之成为新的有序表。在最坏情况下,即初始排序序列是逆序的情况下,比较次数为 $n(n-1)/2$,移动次数为 $n(n-1)/2$ 。

希尔排序法:先将整个待排元素序列分割成若干个子序列(由相隔某个“增量”的元素组成的)分别进行直接插入排序,待整个序列中的元素基本有序(增量足够小)时,再对全体元素进行一次直接插入排序。

c)选择类排序法。

简单选择排序法:扫描整个线性表,从中选择最小的元素,将它交换到表的最前面;然后对剩下的子表采用同样的方法,直到子表空为止。最坏情况下需要比较 $n(n-1)/2$ 次。

堆排序的方法:①将一个无序序列建成堆;②将堆顶元素(序列中的最大项)与堆中最后一个元素交换(最大项应该在序列的最后)。不考虑已经换到最后的那个元素,只考虑前 $n-1$ 个元素构成的子序列,将该子序列调整为堆。反复做步骤②,直到剩下的子序列空为止。在最坏情况下,堆排序法需要比较的次数为 $O(n\log n)$ 。

2.1.2 程序设计基础

(1) 程序设计方法是指设计、编制、调试程序的方法和过程,主要有结构化程序设计方法、软件工程方法和面向对象方法。程序设计风格:良好的设计风格要注重源程序文档化、数据说明方法、语句的结构和输入输出。

(2)结构化程序设计原则

结构化程序设计强调程序设计风格和程序结构的规范化,提倡清晰的结构。a.自顶向下,即先考虑总体,后考虑细节;先考虑全局目标,后考虑局部目标;b.逐步求精是指对复杂问题,应设计一些子目标做过渡,逐步细化;c.模块化是把程序要解决的总目标分解为分目标,再进一步分解为具体的小目标,把每个小目标称为一个模块;d.限制使用 GOTO 语句。

(3)结构化程序基本结构与特点

结构化程序的基本结构有三种类型:顺序结构、选择结构和循环结构。a.顺序结构是指自始至终严格按照程序中语句的先后顺序逐条执行,是最基本、最普遍的结构形式;b.选择结构又称为分支结构,包括简单选择和多分支选择结构;c.重复结构又称为循环结构,根据给定的条件,判断是否需要重复执行某一相同的或类似的程序段。

结构化程序设计中,应注意事项:a.使用程序设计语言中的顺序、选择、循环等有限的控制结构表示程序的控制逻辑;b.选用的控制结构只准许有一个入口和一个出口;c.程序语言组成容易识别的块,每块只有一个入口和一个出口;d.复杂结构应该用嵌套的基本控制结构进行组合嵌套来实现;e.语言中所没有的控制结构,应该采用前后一致的方法来模拟;f.尽量避免 goto 语句的使用。

(4)面向对象的程序设计

面向对象方法的本质是主张从客观世界固有的事物出发来构造系统,强调建立的系统能映射问题域。

对象:通常把对象的操作也称为方法或服务,用来表示客观世界中任何实体,可以是任何有明确边界和意义的东西。对象具有标识唯一性、分类性、多态性、封装性、模块独立性的特征。

类:具有共同属性、共同方法的对象的集合。类是关于对象性质的描述,它同对象一样,包括一组数据属性和在数据上的一组合法操作。

实例:一个具体对象就是其对应分类的一个实例;消息:实例间传递的信息,它统一了数据流和控制流。

消息:消息是实例之间传递的信息,它请求对象执行某一处理或回答某一要求的信息,它统一了数据流和

控制流。一个消息由三部分组成:接收消息的对象名称、消息标识符(消息名)和零个或多个参数。

继承:使用已有的类定义作为基础建立新类的定义技术。继承分为单继承与多重继承。单继承是指,一个类只允许有一个父类,即类等级为树形结构。多重继承是指,一个类允许有多个父类。

多态性:指对象根据所接受的信息而作出动作,同样的信息被不同的对象接收时有不同行动的现象。

面向对象程序设计的优点:与人类习惯的思维方法一致、稳定性好、可重用性好、易于开发大型软件产品、可维护性好。

2.1.3 软件工程基础

(1) 软件功能基本概念

软件是指与计算机系统的操作有关的计算机程序、规程、规则,以及可能有的文件、文档和数据。软件的特点:软件是逻辑实体,有抽象性;生产没有明显的制作过程;运行使用期间不存在磨损、老化问题;开发、运行对计算机系统有依赖性,受计算机系统的限制,导致了软件移植问题;复杂性较高,成本昂贵;开发涉及诸多社会因素。

软件可分为应用软件、系统软件和支撑软件3类。a.应用软件是特定应用领域内专用的软件;b.系统软件居于计算机系统中最靠近硬件的一层,是计算机管理自身资源,提高计算机使用效率并为计算机用户提供各种服务的软件;c.支撑软件介于系统软件和应用软件之间,是支援其它软件的开发与维护的软件。

软件危机指在计算机软件的开发和维护中遇到的一系列严重问题。

软件工程是应用于计算机软件的定义、开发和维护的一整套方法、工具、文档、实践标准和工序,包括软件开发技术和软件工程管理。软件工程包括3个要素:方法、工具和过程。方法是完成软件项目的技术手段;工具支持软件的开发、管理、文档生成;过程支持软件开发的各个环节的控制、管理。

软件生命周期:软件产品从提出、实现、使用维护到停止使用的过程称为软件生命周期。在国家标准中,软件生命周期划分为8个阶段:a.软件定义期包括问题定义、可行性和需求分析3个阶段。b.软件开发期包括概要设计、详细设计、实现和测试4个阶段。c.运行维护期,即运行维护阶段。

软件工程的原则:抽象、信息隐蔽、模块化、局部化、确定性、一致性、完备性和可验证性。

(2) 结构化分析方法

需求分析的任务是发现需求、求精、建模和定义需求的过程,可概括为需求获取、需求分析、编写需求规格说明书和需求评审。常用的分析方法:a.结构化分析方法其实质着眼于数据流,白顶向下,逐层分解,建立系统的处理流程;b.面向对象分析方法。

结构化分析常用工具包括数据流图、数据字典(核心方法)、判断树和判断表。

a.数据流图:又称DFD图,似图形的方式描绘数据在系统中流动和处理的过程,它只反映系统必须完成的逻辑功能,是一种功能模型。符号名称作用:箭头代表数据流,沿箭头方向传送数据的通道;网或椭圆代表加工,输入数据经加工变换产生输出;双杠代表存储文件,表示处理过程中存放各种数据文件;方框代表源和潭,表示系统和环境的接口

b.数据字典:结构化分析的核心。数据字典是对所有与系统相关的数据元素的一个有组织的列表,以及精确的、严格的定义,使得用户和系统分析员对于输入、输出、存储成分和中间计算结果有共同的理解。

c.判定树:使用判定树进行描述时,应先从问题定义的文字描述中分清判定的条件和判定的结论,根据描述材料中的连接词找出判定条件之间的从属关系、并列关系、选择关系,根据它们构造判定树。

d.判定表:与判定树相似,当数据流图中的加工要依赖于多个逻辑条件的取值,即完成该加工的一组动作是由于某一组条件取值的组合引发的,使用判定表比较适宜。

(3) 软件需求规格说明书

软件需求规格说明书是需求分析阶段的最后成果,是软件开发的重要文档之一。

a.软件需求规格说明书的作用:①便于用户、开发人员进行理解和交流;②反映出用户问题的结构,可以作为软件开发工作的基础和依据;③作为确认测试和验收的依据。

b.软件需求规格说明书的内容:①概述;②数据描述;③功能描述;④性能描述;⑤参考文献;⑥附录。

c.软件需求规格说明书的特点:①正确性;②无歧义性;③完整性;④可验证性;⑤一致性;⑥可理解性;⑦可修改性;⑧可追踪性。

(4)结构化设计方法

软件设计是一个把软件需求转换为软件表示的过程。软件设计包括软件结构设计、数据设计、接口设计、过程设计。结构设计定义软件系统各主要部件之间的关系;数据设计将分析时创建的模型转化为数据结构的定义;接口设计是描述软件内部、软件和协作系统之间以及软件与人之间如何通信;过程设计则是把系统结构部件转换为软件的过程性描述。

软件设计基本原理:抽象、模块化、信息隐藏、模块独立性(度量标准:耦合性和内聚性,高耦合、低内聚)。基本思想是将软件设计成由相对独立、单一功能的模块组成的结构。

软件设计分两步完成:概要设计和详细设计。

概要设计的4个任务:设计软件系统结构、数据结构及数据库设计、编写概要设计文档、概要设计文档评审。

面向数据流的设计方法:数据流图的信息分为交换流和事物流,结构形式有交换型和事务型。

详细设计的工具包括:图形工具:程序流程图、N-S、PAD、HIPO;表格工具:判定表;语言工具:PDL(伪码)。

(5)软件测试

测试目的:为了发现错误而执行程序的过程。

测试准则:所有测试应追溯到用户需求;严格执行测试计划,排除测试的随意性;充分注意测试中的群集现象;程序员应避免检查自己的程序;穷举测试不可能;妥善保存设计计划、测试用例、出错统计和最终分析报告。

测试方法:按是否需要执行被测软件的角度,可分为静态测试和动态测试,按功能分为白盒测试和黑盒测试。a.白盒测试是根据程序的内部逻辑设计测试用例,主要方法有逻辑覆盖测试、基本路径测试等;b.黑盒测试是根据规格说明书的功能来设计测试用例,主要诊断方法有等价划分法、边界值分析法、错误推测法、因果图法等,主要用于软件确认测试。

测试实施:软件测试是保证软件质量的重要手段,软件测试是一个过程,其测试流程是该过程规定的程序,目的是使软件测试工作系统化。

软件测试过程分4个步骤,即单元测试、集成测试、验收测试和系统测试。

a.单元测试是对软件设计的最小单位——模块(程序单元)进行正确性检验测试。单元测试的目的是发现各模块内部可能存在的各种错误。单元测试的依据是详细的设计说明书和源程序。单元测试的技术可以采用静态分析和动态测试。

b.集成测试是测试和组装软件的过程,主要目的是发现与接口有关的错误,主要依据是概要设计说明书。集成测试所设计的内容包括:软件单元的接口测试、全局数据结构测试、边界条件和非法输入的测试等。集成测试时将模块组装成程序,通常采用两种方式:非增量方式组装和增量方式组装。

c.确认测试的任务是验证软件的功能和性能,以及其他特性是否满足了需求规格说明中确定的各种需求,包括软件配置是否完全、正确。确认测试的实施首先运用黑盒测试方法,对软件进行有效性测试,即验证被测软件是否满足需求规格说明确认的标准。

d.系统测试是通过测试确认的软件,作为整个基于计算机系统的一个元素,与计算机硬件、外设、支撑软件、数据和人员等其他系统元素组合在一起,在实际运行(使用)环境下对计算机系统的一系列集成测试和确认测试。系统测试的具体实施一般包括:功能测试、性能测试、操作测试、配置测试、外部接口测试、安全性测试等。

(6)程序调试

调试任务:诊断和改正程序中的错误。

调试主要在开发阶段进行。程序调试活动由两部分组成,一是根据错误的迹象确定程序中错误的确切性质、原因和位置;二是对程序进行修改,排除这个错误。

程序调试的基本步骤:a)错误定位,从错误的外部表现形式入手,研究有关部分的程序,确定程序中出错位置,找出错误的内在原因;b)修改设计和代码,以排除错误;c)进行回归测试,防止引进新的错误。

软件调试可分为静态调试和动态调试。静态调试主要是指通过人的思维来分析源程序代码和排错,是主要的设计手段,而动态调试是辅助静态调试的。主要的调试方法:强行排错法、回溯法和原因排除法。

► 3.1 C 语言概述

3.1.1 C 语言的特点

- (1)语言简洁、紧凑,并且使用方便、灵活;
- (2)运算符丰富;
- (3)数据结构丰富;
- (4)具有结构化的控制语句;
- (5)语法限制不太严格,使程序设计比较自由;
- (6)C 语言允许用户直接访问物理地址,能进行位(bit)操作,可以直接对硬件进行操作。

3.1.2 C 语言的书写格式

C 语言书写格式自由,一行内可以写几个语句,一个语句也可以分写在多行上。C 程序没有行号,每个语句和数据定义的最后必须有一个分号。C 语言中分号是语句中不可少的,即使是程序中的最后一个语句也应该包含分号。C 语言中的注释可以用“/*”开始,用“*/”结束,注释可以在任何允许插入空格的地方插入。C 语言中注释不允许嵌套,注释可以用西文,也可以用中文。

3.1.3 C 语言程序文件

用 C 语言编写的程序称为 C 语言源程序,源程序文件的后缀名为“.c”。源程序经编译后生成后缀名为“.obj”的目标文件,把目标文件与各种库函数连接起来,生成“.exe”可执行文件。

3.1.4 C 语言的函数

一个 C 源程序实际上就是若干函数的集合,这些函数中有一个是程序的主函数,任何 C 的源程序执行时,都是从主函数开始执行的,其它的函数最终必将被这个主函数所调用。C 语言除了主函数规定必须取名 main 外,其它的函数名可以任取,但是要符合 C 的标识符取名规则,另外注意不要与保留字重名,最好也不可 C 语言中的库函数或其它一些命令如编译预处理命令重名。各个函数在程序中所处的位置并不是固定的,但要求一个函数是完整的、独立的。不允许出现在一个函数内部又去定义另一个函数,或是函数格式不齐全的现象。

一个完整的 C 源程序的格式可以如下表示:

```
编译预处理  
main 函数()  
子函数 1()  
子函数 2()  
子函数 3()
```

main 函数又称主函数,是 C 程序的入口。main 后面跟一对小括号()和一对花括号{|,小括号中的内容为函数的参数,可以为空;花括号括起来的部分称为 main 函数的函数体。

一个 C 程序从 main 函数开始执行,到 main 函数体执行完结束,而不论 main 函数在整个程序中的位置如何。每一个程序有且仅有一个 main 函数,其他函数都是为 main 函数服务的。

3.1.5 C 语言的基本词法

C 语言的基本词法由三部分组成:符号集、关键字、保留字。

符号集就是一门语言中允许出现的字符的集合,C 语言的符号集就是 ASC II 码表中的一些字符,在键盘上不能直接得到,C 语言引入了转义字符的概念,利用反斜杠符号“\”后加上字母的一个字符组合来表示这些字

符,当在源程序中遇到这类字符组合时,虽然这个字符组合是一个字符串的形式,但 C 语言仍会自动将之理解成某一特定的字符。比如“\n”,C 语言在处理这个字符组合时,会自动理解成回车换行符号。转义字符经过进一步引申应用,形成了另外两种形式:“\ddd”和“\xnn”,这里“\”后的 ddd 和 xnn 分别代表三位八进制和两位十六进制数(打头的“x”只是标明后面跟着的是十六进制数),这两种形式不再局限于表示不可打印的字符,它们可以表示 ASC II 码表中的任意字符,只要把所需表示的字符的 ASCII 码转换成八进制数或十六进制数即可。比如说字母“A”,ASCII 码为 65,65 的八进制和十六进制分别为 101 和 x41,所以,字母 A 可表示为“\101”或“\x41”,对转义字符应认真理解。

标识符就是用以标识的符号。正如现实生活中给每一个人都取一个名字一样,C 语言中的每一个对象(如函数、变量等)都必须取一个标识符以和其它对象区别开。在 C 语言中,这个标识符是一个字符串,这个字符串的选定有一定的规则;必须是以字母或下划线开头的字母与数字的序列。

关键字实际上就是一些特殊的标识符,又称保留字,这些保留字不允许用户对它重新定义。

C 语言中的关键字共有 37 个,即 auto、break、case、char、const、continue、default、do、double、else、enum、extern、float、for、goto、if、inline、int、long、register、restrict、return、short、signed、sizeof、static、struct、switch、typedef、union、unsigned、void、volatile、while、_bool、_Complex、_Imaginary。

3.1.6 存储形式

在计算机中保存数据是采用二进制形式,由 0 或 1 构成的二进制称为位(bit),8 个位构成一个字节(Byte),即 1 个 Byte 由 8 个 bit 构成。

二进制、八进制、十六进制转化为十进制采用乘法转换,十进制转化为二进制、八进制、十六进制采用除法转换。数据的存放位置就是它的地址。

3.1.7 注释

注释是对程序的说明,可出现在程序中任意合适的地方,注释有两种方式,即行注释“//”和块注释“/*...*/”。

行注释从注释开始,到行尾结束;块注释从“/*”开始到最近一个“*/”结束,其间任何内容都不会被计算机执行;注释不可以嵌套。

3.1.8 算法

做任何事情都有一定的步骤。为解决一个问题而采取的方法和步骤,就称为算法。计算机算法:计算机能够执行的算法。

计算机算法可分为两大类:

- (1)数值运算算法用于求解数值;
- (2)非数值运算算法用于事务管理领域。

3.1.9 算法的特性

有穷性:一个算法应包含有限的操作步骤而不能是无限的;

确定性:算法中每一个步骤应当是确定的,不能是含糊、模棱两可的。输入:有零个或多个输入;

输出:有一个或多个输出;

有效性:算法中每一个步骤应当能有效地执行,并得到确定的结果。

3.1.10 头文件

也称为包含文件或标题文件,一般放在一个 C 语言程序的开头,用 #include“文件名”或<文件名>的格式,其中文件名是头文件名,一般用.h 作为扩展名。

▶ 3.2 数据类型、运算符与表达式

3.2.1 标识符的规则

标识符就是标识名字的有效字符序列,可以理解为 C 程序中的单词。其命名规则为:

- (1)只能由字母、数字和下划线组成,字母区分大小写;
- (2)第一个字符必须是字母或下划线,不能为数字。

3.2.2 C 语言标识符的分类

- (1)关键字。它们在程序中有固定的含义,不能另作他用,如 int、for、switch 等。
- (2)预定义标识符。预先定义并具有特定含义的标识符,如 define、include 等。
- (3)用户标识符。用户根据需要定义的标识符,符合命名规则且不与关键字相同。

3.2.3 常量与变量

常量是指在程序运行过程中,其值不能改变的量。常量分为整型常量、实型常量、字符常量、字符串常量和符号常量 5 种。

变量是指在程序运行过程中,其值可以改变的量。C 语言中没有字符串变量。存放字符串使用字符数组或字符指针。

3.2.4 整型数据

整型变量可分为基本整型(int)、短整型(short)、长整型(long)、双长整型(long long)。根据是否需要符号位,又分为有符号整型(signed)或无符号整型(unsigned),可以扩展为 8 种组合形式。

整型常量有十进制、八进制、十六进制三种表示形式,没有二进制形式。八进制整型常量加前导数字 0,后面由 0~7 构成;十六进制常量加前导 0X 或 0x,后面由 0~9、A~F(或 a~f)构成。

不同的计算机对上述几种整型数据所占用的内存字节数和数值范围有不同的规定,以 IBM-PC 微机为例,在 VC++6.0 中,编译系统为 int 型数据分配 4 个字节。

3.2.5 实型数据

C 语言中的实型常量有两种表示形式:十进制数形式和指数形式。在用指数形式表示实型数据时,字母 E 可以用小写 e 代替,指数部分必须是整数(若为正整数时,可以省略“+”号)。

C 语言中的实型变量分为两种:单精度类型和双精度类型,分别用保留关键字 float 和 double 进行定义。在一般系统中,一个 float 型数据在内存中占 4 个字节;一个 double 型数据占 8 个字节(一个 long double 型数据占 16 个字节)。

3.2.6 字符型数据

C 语言的字符常量代表 ASC II 码字符集里的一个字符,在程序中要单引号括起来。C 语言规定字符常量可以作为整数常量来处理(注:这里的整数常量指的是相应字符的 ASC II 代码,因此字符常量可以参与算术运算)。在 C 语言中还有一类特殊形式的字符常量,称为“转义字符”。这类字符常量是以一个反斜杠开头的字符序列,但它们只代表某个特定的 ASC II 码字符,在程序中使用这种常量时要括在一对单引号中。

字符变量:C 语言中的字符变量用关键字 char 来定义,每个字符变量中只能存放一个字符。在一般系统中,一个字符变量在计算机内存中占一个字节。与字符常量一样,字符变量也可以出现在任何允许整型变量参与的运算中。

字符串常量:C 语言中的字符串常量是由一对双引号括起来的字符序列。注意不要将字符常量和字符串常量混淆。C 语言对字符串常量的长度不加限制,C 编译程序总是自动地在字符串的结尾加一个转义字符‘\0’,作为字符串常量的结束标志。C 语言中没有专门的字符串变量,如果要把字符串存放在变量中,则要用一个字符型数组来实现。

3.2.7 运算符

C 语言中的运算符可以归纳下列 5 类:算术运算符、关系运算符、赋值运算符、逻辑运算符和条件运算符。

算术运算符:算术运算符有+、-、*、/、%共 5 种。求余运算符“%”要求运算对象均为整型数据,其他运算符的运算对象可以为任意数据类型。算术表达式的求值规律与数学中的四则运算规律类似,其运算规律和要求为:

在算术表达式中,可使用多层括号,但左右括号必须配对。运算时从内层圆括号开始,由内向外依次计算表达式的值。在算术表达式中,若包含不同优先级的运算符,则按运算符的优先级别由高到低进行,若表达式中运算符的级别相同,则按运算符的结合方向进行。

关系运算符:在算术表达式中,C语言提供6种关系运算符:<,>,<=,>=,==,!=。前四种运算符(<,>,<=,>=)的优先级相同,后两种的优先级也相同,并且前四种的优先级高于后两种。关系运算符属于双目运算符,其结合方向为自左至右。用关系运算符可以将两个表达式(包括算术表达式、关系表达式、逻辑表达式、赋值表达式和字符表达式)连接起来构成关系表达式。

赋值运算符:赋值运算符为“=”,赋值运算是把赋值运算符右边表达式的值赋给左边变量。所以,赋值运算符左边必须为变量或存储单元,也不同于关系运算中的等于运算符“=”。

赋值表达式格式为:

变量名=表达式

复合赋值运算符是将算术运算符或位运算符与赋值运算符组合在一起组成的运算符,常见的有+=、-=、*=、/=、%=。掌握复合赋值表达式转化为赋值表达式的方法。

逻辑运算符:C语言提供三种逻辑运算符:&&(逻辑与)、|| (逻辑或)、!(逻辑非)。其中前两种为双目运算符,第三种是单目运算符。

条件运算符:C语言中把“?:”称作条件运算符。条件运算符要求有三个运算对象,它是C语言中唯一的一个三目运算符。由条件运算符构成的条件表达式的一般形式为:

表达式1?表达式2:表达式3

当表达式1的值为非零时,取表达式2的值为条件表达式的值;当表达式1的值为零时,取表达式3的值为条件表达式的值。

条件运算符具有自右向左的结合性,其优先级别比关系运算符和算术运算符都低。

3.2.8 不同类型数据间的转换与运算

将一个运算对象转换成指定的数据类型,格式为:(类型名)表达式

3.2.9 自增自减运算

自增运算符“++”与自减运算符“--”是单目运算符,运算对象必须是变量。自增自减运算分前缀运算和后缀运算,它们所对应的表达式的值是有区别的。例如,j=i++等价于j=i,i++;而j=++i;等价于i=i+1;j=i;。

3.2.10 逗号运算

逗号运算符运算优先级最低,可将多个表达式构成一个新的表达式。整个逗号表达式的值等于最后一个表达式的值。

► 3.3 基本语句

3.3.1 C语言语句

C语言的语句共分五大类:表达式语句、控制语句、函数调用语句、空语句和复合语句。

表达式语句的一般形式为:

表达式;

最典型的表达式语句是由一个赋值表达式加一个分号构成的赋值语句,如int x; x=10; float y; y=3.14;。

控制语句是C语言程序设计中用来构成分支结构和循环结构的语句。此类语句有if语句,for语句,while语句,do-while语句,switch语句等。

函数调用语句的一般形式为:

函数名(实参表);

空语句的一般形式为

;

这条语句的含义是什么也不做。凡是在 C 语句程序中出现语句的地方都可以用一个分号来代替一条语句。复合语句的一般形式为

```
|语句 1;语句 2;...;|
```

复合语句在功能上相当于一语句。

3.3.2 数据输入与输出

C 语言本身没有提供输入、输出操作语句,C 程序的输入和输出完全依靠调用 C 语言的标准输入、输出函数来完成。常用的输入、输出函数是:

printf 函数、scanf 函数、putchar 函数、getchar 函数、gets 函数、puts 函数。

3.3.3 printf 函数

printf 函数是 C 语言提供的标准输出函数,它的作用是在终端设备(或系统隐含指定的输出设备)上按指定格式进行输出。printf 函数的一般调用格式为:

```
printf("格式控制",输出列表);
```

格式控制是用一对双引号括起来的,包含格式说明和原样信息。输出列表包含若干输出项。格式控制参数以字符串的形式描述,由两部分组成。

(1)普通字符:将被简单地显示。

(2)格式字符:将引起一个输出参数项的转换和显示,由%引出并以一个类型描述符结束的字符串,中间可加一些可选的附加说明项。

3.3.4 printf 函数中格式说明

%d:表示数据以十进制整型输出。

%o:表示数据以八进制无符号整型输出。

%x:表示数据以十六进制整型输出。

%u:表示数据以十进制无符号整型输出。

%f:表示数据以单精度实型输出。

%e:表示数据以指数形式输出。

%c:表示数据是以字符型输出。

%s:表示数据以字符串的形式输出。

可在"%"和格式字符之间加一个数来控制数据所占的宽度和小数位数。

3.3.5 scanf 函数

scanf 函数是 C 语言提供的标准输入函数,它的作用是在终端设备(或系统隐含指定的输入设备)上输入数据。

scanf 函数的一般调用格式为:

```
scanf("格式控制",输入列表)
```

格式控制是用双引号括起来的字符串,也称格式控制串。格式控制串的作用是指定输入时的数据转换格式,即格式转换说明。格式转换说明由"%"符号开始,其后是格式描述符。

输入列表中的各输入项用逗号隔开,各输入项只能是合法的地址表达式,即在变量前加一个地址符号。

当用键盘输入多个数据时,数据之间可用分隔符隔开。分隔符包括空格符、制表符和回车符。

3.3.6 getchar 函数

getchar 函数的作用是从标准输入设备(通常指键盘)上读入一个字符。一般调用形式为:

```
getchar();
```

getchar 函数本身没有参数,其函数值就是从输入设备得到的字符。