

# 广东培正学院 2016 年本科插班生专业课考试大纲

## 《程序设计基础》考试大纲

### I. 考试性质

《程序设计基础》是信息技术（IT）类专业的一门学科核心基础课，是培养学生掌握程序设计思维、理念、方法的入门课程，其代表语言为 C 语言。

熟悉程序设计思维和理念，掌握 C 语言程序设计的最基本的知识、技术和方法，具备最基本的程序设计能力是专科学生进入本科学习必备要素，也是进一步学习其他程序设计课程和进行软件开发的基础。

本课程考试在考核上述必备要素的基础上，选拔成绩合格的专科生进入学校本科高年级阶段修读。符合条件者还将获得计算机科学与技术专业工学学士学位。

### II. 考试内容及要求

#### 一 考试基本要求

《程序设计基础》课程考试要求学生初步掌握 C 语言程序设计的知识和技能，即 C 语言程序设计的基本概念；C 程序的基本结构；C 语言的语法、句法；能熟练地阅读、理解简短的 C 语言程序；会分析一般性、典型的应用问题，确定其算法并编写相应程序。

#### 二、考核知识点及考核要求

本大纲的考核要求分为“识记”、“领会”、“应用”三个层次，具体含义是：

识记：能正确认识和表达名词，能解释有关的概念、知识的含义。

领会：在识记的基础上，能掌握基本原理、基本方法，能掌握有关概念、原理、方法的区别与联系。

应用：在理解的基础上，能运用基本概念、基本理论、基本方法分析和解决一般性理论问题和应用问题。

### 专题一 C 语言程序设计及算法

#### 一、考核知识点

##### 1、C 语言及其特点

- 2、C 程序结构
- 3、C 程序的编译与运行
- 4、程序=算法（对操作的描述）+ 数据结构（对数据的描述）
- 5、算法的三种表示方法：自然语言法、流程图法、伪代码法。

## 二、考核要求

### 1、识记

- (1) C 语言及其特点
- (2) 程序=算法（对操作的描述）+ 数据结构（对数据的描述）
- (3) 结构化程序的三种基本结构：顺序结构、选择结构、循环结构
- (4) 认识流程图的各种框图

### 2、领会

- (1) 算法的意义
- (2) 流程图表示算法的思路
- (3) 结构化程序的三种基本结构的区别

### 3、应用

- (1) 书写最简单 C 语言程序
- (2) 会进行 C 程序的编辑、编译与运行

## 专题二 数据类型、运算符和表达式

### 一、考核知识点

- 1、C 语言字符集
- 2、常量与变量的表示和书写
- 3、常量与变量的取值范围
- 4、运算符的书写和意义
- 5、表达式的构成
- 6、表达式求值

### 二、考核要求

#### 1、识记

- (1) 常用数据类型：整型、字符型、实型、数组、指针的认识；变量及标识符的命名

规则

- (2) 整型、字符型、单精度实型数据的取值范围
- (3) 算术运算符；赋值运算符；逗号运算符的认知及其运算法则
- (4) 字符常量；英文字符的 ASCII 码；转义字符；字符变量的定义
- (5) 自增、自减运算符认知和运算法则
- (6) C 语言表达式的意义和构成

## 2、领会

- (1) 赋值运算符“=”的操作过程
- (2) 字符与字符串的区别，字符与字符串的赋值方法
- (3) 数据的取值范围及变量值溢出
- (4) 赋值运算“=”与值相等“==”的区别；如何交换两个变量的值
- (5) 自增运算、自减运算的前置及后置的区别
- (6) 算术表达式运算顺序
- (7) 混合运算时如何确定表达式值得类型

## 3、应用

- (1) 能根据数据类型正确定义变量
- (2) 会定义字符和字符串
- (3) 能正确书写算术表达式
- (4) 会进行赋值、算术、逗号、自增和自减运算
- (5) 掌握赋值数据类型转换规则，会对一般算术表达式和简单逗号表达式求值

## 专题三 顺序程序设计和格式化输入、输出

### 一、考核知识点

- 1、C 语句书写规则和分类
- 2、格式输出函数 printf 认识和作用
- 3、格式输入函数 scanf 认识和作用，“&”符号的意义
- 4、顺序程序执行过程
- 5、字符输入、输出函数

### 二、考核要求

#### 1、识记

- (1) printf 及其格式控制符%d、%ld、%c、%f 认识，输出表列中表项的计算顺序（自

右向左。)

(2) scanf 及其格式控制符%d、%c、%f、%s; 认识, 键盘输入时数据的分隔, 控制字符串中的非格式字符

## 2、领会

(1) “&”符号的意义

(2) printf 及其格式控制符%d、%ld、%c、%f 的作用

(3) scanf 及其格式控制符%d、%c、%f、%s 的作用

## 3、应用

(1) 会使用 printf 函数输出数据

(2) 会使用 scanf 函数输出数据

(3) 会使用 putchar 函数 和 getchar 函数输出、输入字符

# 专题四 选择结构程序与循环结构程序

## 一、考核知识点

1、关系运算符和表达式

2、逻辑运算符和逻辑表达式

3、if 语句和 if...else...语句

4、if 语句嵌套

5、switch 多分支选择结构和 break 语句

6、条件运算符和条件表达式

7、循环结构程序的构成,

8、while 循环

9、do ..... while 循环

10、for 循环, for 循环中的三个表达式

11、二重循环

12、循环的中断退出 (break) 和本次循环的提前结束(continue)

## 二、考核要求

### 1、识记

(1) 六种关系运算符及其运算优先级

(2) 逻辑运算符 (&& 、||、!) 的理解, 逻辑表达式的书写

(3) 条件运算符( ? : )

(4) 逻辑表达式的值:

系统在输出逻辑表达式值时, 以数值 1 代表“真”, 以 0 代表“假”

而在输入一个值让系统判断“真”、“假”时, 以 0 代表“假”, 以非 0 代表“真”

(5) if 语句的书写格式

(6) while、do—while 、 for 语句的书写格式

(7) for 循环中三个表达式 (循环变量赋值表达式, 循环条件表达式, 循环变量增值表达式) 的构成规则

## 2、领会

(1) 赋值运算符=和关系运算符==的区分

(2) 逻辑运算与算术运算的区别

(3) 逻辑表达式的求值过程

(4) 条件表达式的求值过程 (自右向左); 如:  $a > b ? a : c > d ? c : d$  相当于:  $a > b ? a : (c > d ? c : d)$

(5) if 语句

单选择 if 语句 “if(表达式) 语句;” 的执行过程

双选择 if 语句 “if(表达式) 语句 1; else 语句 2;” 的执行过程, 程序分支后的汇

合点

(6) 对称 if 语句与条件表达式的互换, 如:  $\text{if}(a > b) \text{ max} = a;$

$\text{else max} = b; \longleftrightarrow \text{max} = (a > b) ? a : b$

(7) 嵌套 if 语句中 else 与 if 的配对规则

(8) switch 结构和 break 语句, switch 结构执行过程

(9) 条件循环 (while 循环和 do while 循环) 执行过程, 两种循环的区别

(10) 计数循环 (for 循环)

for 循环的执行过程及循环次数; for 循环中三个表达式 (循环变量赋值表达式, 循环条件表达式, 循环变量增值表达式) 的构成规则与注意事项

(11) 死循环发生的条件, 如何避免死循环

(12) 循环的中断退出 (break) 和本次循环的提前结束(continue)的执行路线

## 3、应用

(1) 会求逻辑表达式的值

(2) 依据汉语表述书写逻辑表达式, 例如: 闰年的判断、判断一个字符是大写或小写

字母。

(3) 单选择 if 语句结构程序和双选择 if 语句结构程序的阅读及编写

(4) 条件循环（while 循环和 do while 循环）和计数循环（for 循环）程序的阅读及编写

(5) 二重循环结构程序阅读和编写

(6) 典型应用程序例

## 专题五 函数

### 一、考核知识点

- 1、程序模块与函数
- 2、函数的定义与分类
- 3、函数的调用与返回
- 4、被调用函数的声明和函数原型
- 5、形式参数和实际参数
- 6、函数的嵌套调用
- 7、数组元素和数组名作为函数参数
- 8、局部变量和全局变量
- 9、变量的作用域和生存期
- 10、变量的声明和定义
- 11、内部函数和外部函数

### 二、考核要求

- 1、识记
  - (1) C 程序的构成，main() 函数
  - (2) 函数定义与被调函数的声明
  - (3) 标准函数（库函数）与用户自定义函数
  - (4) 函数值与函数的参数
  - (5) 形参和实参的定义，实参和形参在数量上，类型上，顺序上应严格一致。
  - (6) 局部变量和全局变量的定义
  - (7) 变量的作用域和生存期
  - (8) 变量的声明

(9) 内部函数和外部函数定义

## 2、领会

(1) 多函数构成的 C 程序的执行过程

(2) 变量为什么要声明？会声明局部变量和全局变量

(3) 为什么要声明被调函数？如何声明被调函数？

(4) 函数的调用与返回的执行过程

(5) 形参和实参如何结合并传递数值？

(6) 函数的返回值的类型

(7) 函数调用的三种形式：

函数语句（无函数值返回）；

函数表达式（被调函数作为表达式中的一项参加运算）；

函数实参（被调函数作为另一函数实参）；

(8) 数组元素和数组名作为函数参数时的“传值”和“传址”，“传值”的单向性和“传址”时实现数值传递的双向性。

(9) 理解变量的作用域和生存期

## 3、应用

(1) 能正确定义函数、定义形参和实参

(2) 会编写多函数构成的程序

(3) 熟悉典型程序例

# 专题六 数组和指针

## 一、考核知识点

1、一维数组定义和引用

2、二维数组定义和引用

3、字符数组

4、指针即是变量在内存中的首地址

5、指针变量

6、通过指针引用数组

7、通过指针引用字符串

8、指向函数的指针

## 9、返回指针值的函数

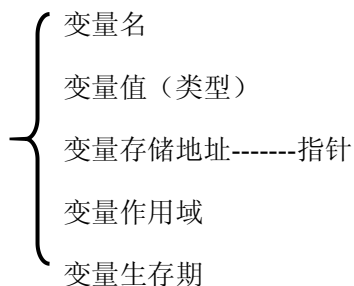
## 10、指针数组

### 二、考核要求

#### 1、识记

- (1) 数组的定义：数组是一组同类型数据的有序集合
- (2) 一维数组的定义
- (3) 二维数组的定义
- (4) 字符数组的定义，字符数组的初始化
- (5) 字符串的赋值，字符串的结束标志
- (6) 字符串函数：strcat、strcpy、strcmp、strlen、strlwr、strupr
- (7) 变量的属性

一个变量有如下几个属性：



- ① 指针其实就是变量的地址。
  - ② 地址也是一种数据类型，它可以存放在一种特殊的变量-----“指针变量”中
  - ③ 定义指针的目的是为了通过指针去访问变量（存取变量的值）。
- (8) 指针和指针变量区别
  - (9) 指针运算符 “\*”
  - (10) 取变量地址运算符 “&”
  - (11) 指针变量的定义

指针变量的定义形式：例如：

```
int *p2;           /* p2 是指向整型变量的指针变量 */
float *p3, x;     /* p3 是指向浮点变量的指针变量 */
char *p4;        /* p4 是指向字符变量的指针变量 */
```

定义包括三个内容：●定义变量为一个指针变量

●指针变量的名称



## ● 指针所指向的变量的数据类型

(12) 由指针作为数组元素构成的指针数组

### 2、领会

- (1) 一维数组名中存放数组的首地址，一维数组的长度，下标的起止范围
- (2) 二维数组名中存放数组的首地址，二维数组元素的输入与输出、下标的起止范围
- (3) 数组名与指针的关系
- (4) 二维数组的行指针
- (5) 字符串的实际长度与存储长度，字符串函数的一些使用规则：

例如：if(字符串 1 和字符串 2 相同) 返回函数值为 0。

If(字符串 1>字符串 2) 返回函数值为正数,其值是 ASCII 码的差值。

If(字符串 1<字符串 2) 返回函数值为负数,其值也是 ASCII 码的差值。

不能这样比较二个字符串：

```
if(str1==str2) printf("yes");
```

只能用： `if(strcmp(str1,str2)==0) printf("yes");`

(6) 变量的访问

#### ① 直接访问方式

按变量地址(程序编译后，变量名就成为了内存地址)存取变量值的方式称为直接访问方式。如语句：`printf("%d",i); scanf("%d",&i);`

#### ② 间接访问方式

假设已经用 `int *p;` 定义了指针变量 `p`，其中存放着整型变量 `i` 的地址(如：2000H)。而指针变量本身被分配到内存单元 3010H 和 3011H 存放。

通过执行语句“`p=&i;`”将整型变量 `i` 的地址存放到指针变量 `p` 中，即 `p` 的值为变量 `i` 所对应的内存单元的起始地址 2000H。现在要存取 `i` 的值可以这样做：

找到存放 `i` 的地址的内存单元地址(3010H 和 3011H)，从中取出变量 `i` 的地址值 2000H，对 2000H 和 2001H 单元进行存取——这就称为间接访问。

(7) 变量的指针和指向指针的指针变量的区别

#### ① 指针运算符 “\*” (间接访问变量的值)

用 “\*” 表示指针变量和它所指向的变量之间的联系。

如有：`int *ip, i;`

```
ip=&i; /*指针变量 ip 中存放整型变量 i 的地址*/
```

则：`i=5;` 与 `*ip=5;` 两个语句作用相同。

结论：`*ip` 与 `i` 等价。所以有：“`x=i+3;`”与“`x=*ip+3;`”等价。

## ② 取变量地址运算符 “&”

取地址运算符 “&” 与一个变量连用，可以得到该变量的内存地址。如：`&a` 即是变量 `a` 的地址。

## (8) 数组名和指针作为函数的参数

由于被调函数中形参接收到的值是指针（实参的地址），于是形参和实参同时指向若干内存单元，于是在被调函数中就可以引用这些地址而修改其内存单元的值，从而达到修改主调函数中相应变量值的目的。

## 3、应用

(1) 会在程序中引用一维数组及其数组元素

(2) 会在程序中引用二维数组及其数组元素

(3) 会在程序中输入和输出字符串，会使用字符串函数编程

(4) 指针变量的赋值和引用

### ① 指针变量的赋值

● 指针变量在使用之前必须赋值。

● 指针变量只能赋予变量的地址值。

可以用以下两种方式给指针赋值：

● 用指针变量初始化的方法，即：`int a; int *p=&a;`

● 用赋值语句的方法：`int a; int *p; p=&a;`

### ② 指针变量的引用

表达式中的 `*p` 与变量定义中的 `*p` 含义不同，如有以下语句：

```
int i, j, *p=&i; /*定义一个指针*/
```

```
*p=5;          /*给变量 i 赋值 */
```

```
j=*p          /*通过指针 P 引用变量 i */
```

(5) 会在程序中使用指针引用数组、字符串

(6) 能在程序中使用数组和指针作为函数的参数

## 专题七 文件

### 一、考核知识点

1、什么是文件？程序文件和数据文件

2、文件的相关知识：

文件名；文本文件和二进制文件；文件缓冲区；文件类型指针

3、打开与关闭文件

4、顺序读写数据文件

5、随机读写数据文件

6、文件读写的出错检测

## 二、考核要求

1、识记

(1) 什么是文件

文件是存储在外部介质（磁介质、光介质）上的数据集合。只有在使用时才调入内存中来。

(2) 文件分类

程序文件

数据文件

(3) C 语言的数据文件是流式文件：

C 语言系统把它们都看成是一连串的字符，即字符流(stream)。对文件的存取以字符（字节）为单位，按字节进行处理。输入输出字符流的开始和结束只由程序控制而不受物理符号(如回车符)的控制。

C 语言的数据文件属于“缓冲区文件系统”。数据读写都要经由内存“缓冲区”来进行。

(4) 文件类型指针

对于每个被使用的文件，系统都在内存中建立一个名为 FILE 的“结构体”，用来存放该文件的有关信息，（文件名、缓冲读写缓冲区的大小、文件状态、读写位置等），为了方便地获取文件的这些信息，定义一个“指针变量”来访问 FILE 结构体，这个指针变量称之为“文件指针”，一个文件对应一个文件指针。

2、领会

(1) 文件目录（文件夹）和路径，文件主名和扩展名

(2) C 语言文件的读写是字节为单位进行的

(3) 文件类型指针和文件内部读写指针的区别

文件指针是指向整个文件的，须在程序中定义说明，只要不重新赋值，文件指针的

值是不变的。文件内部的读写指针用以指示文件内部的当前读写位置，每读写一次，该指针均要发生移动，其值由系统自动改变的。

(4) 文件为什么要及时关闭?

(5) 顺序文件和随机文件的区别

### 3、应用

(1) 认知文件操作函数并熟悉其功能

(2) 简单顺序文件的建立、打开、读写和关闭

(3) 简单随机文件的建立、打开、读写和关闭

## III. 考试形式及试卷结构

试卷特点是难度不大但题量充足，考查最基本的知识和能力。

1、考试形式为闭卷，笔试，考试时间为 120 分钟，试卷满分为 100 分。

2、试卷内容比例：第 1~2 专题约占 25%，第 3~5 专题约占 55%，第 6~7 专题约占 20%，

3、试卷题型比例：客观题 40 分（单项选择题 30%、判断题 10%）；主观题 60 分，（填空题 20%、程序阅读题 20%、编程题 20%）。

4、试卷难度比例：易、中、难大致分别为 30%、50%、20%。

## IV. 参考书目

《C 程序设计（第四版）》，谭浩强著，清华大学出版社，2010 年 6 月

《C 程序设计（第四版）学习辅导》，谭浩强编著，清华大学出版社，2010 年 7 月

## V. 题型示例

一、单项选择题（从各题的 4 个备选答案中选出一个正确答案，将其代号写在题干空白处。选错或未选者不得分，每小题 1 分，共 30 分。）

1、用户用 C 语言编写的程序称为（ ）。

A. C 语言执行程序 B. C 语言源程序 C. C 语言目标程序 D. C 语言编译程序

二、判断题（你认为下列命题是正确的，就在其题号前的方框中加“√”，错误的加“×”。每小题 1 分共 10 分。）

