

2021 年广州商学院普通专升本考试大纲

《程序设计基础》考试大纲

I. 考试性质

普通高等学校专升本招生考试（以下简称普通专升本）是由原本科插班生招生考试进行优化调整而来，是专科毕业生参加的选拔性考试。普通高等学校根据考试的成绩，按已确定的招生计划，德智体全面衡量，择优录取。因此，普通专升本考试应有较高的信度、较高的效度、必要的区分度和适当的难度。

本考试科目《程序设计基础》含《C 语言程序设计》和《数据结构》两门课程。《C 语言程序设计》和《数据结构》是数据科学与大数据技术专业 and 智能科学与技术专业的必修课程。

II. 考试形式及试卷结构

一、考试方法

考试形式：闭卷，笔试。考试时间 150 分钟，满分 200 分。

二、考试内容比例

根据考核的要求，适当安排各知识点，达到考核考生对知识点的识记、理解和应用的水平和能力。各知识内容比例为：识记（30%），理解（45%），应用（25%）。

《C 语言程序设计》和《数据结构》两门课程的比例为 8:2。

三、试题类型及比例

试卷的题目类型有：单项选择题（40%）、填空题（20%）、判断题（10%）、程序分析题（15%）、算法设计题（15%）（各种题型的具体样式可参见题型示例）。

四、试题难度比例

难：10%，中等：60%，容易：30%。

III. 考核内容和要求

一、C 语言程序设计部分（160 分）

第一章 引论

不考核。

第二章 用 C 语言编写程序

1. 考核知识点：

常量与变量、基本数据类型、赋值运算符、算术运算符和表达式。

2. 考核要求：



理解：常量与变量、基本数据类型和赋值运算符、算术运算符和表达式。

第三章 分支结构

1.考核知识点：

- (1) 关系运算符和关系表达式、逻辑运算符和逻辑表达式。
- (2) if...else...分支语句、switch 多分支语句。

2.考核要求：

- (1) 识记：关系运算符和逻辑运算符。
- (2) 理解：关系表达式、逻辑表达式、switch 多分支语句。
- (3) 应用：if...else...分支语句。

第四章 循环结构

1.考核知识点：

循环语句、嵌套循环、流程的转移控制。

2.考核要求：

- (1) 识记：循环语句的各个要素。
- (2) 理解：循环语句、嵌套循环、流程的转移控制。
- (3) 应用：循环语句、嵌套循环、流程的转移控制。

第五章 函数

1.考核知识点

函数的定义、调用，函数参数、变量作用域，常用的标准库函数。

2.考核要求

- (1) 理解：函数的定义、调用，函数参数、变量作用域和常用的标准库函数。
- (2) 应用：函数的定义、调用，函数参数、变量作用域和常用的标准库函数。

第六章 回顾数据类型和表达式

1. 考核知识点

数据的输入输出、数据类型自动转换、表达式。

2.考核要求

- (1) 识记：输入、输出的概念。
- (2) 理解：数据的输入输出、数据类型自动转换。
- (3) 应用：表达式。

第七章 数组

1.考核知识点

- (1) 数组的定义、引用、初始化。



- (2) 向函数传递数组。
- (3) 字符串的存储-访问和输入/输出。

2.考核要求

- (1) 识记：数组的概念。
- (2) 理解：数组的定义-初始化和引用、向函数传递数组。
- (3) 应用：字符串的存储-访问和输入/输出。

第八章 指针

1.考核知识点

- (1) 变量的内存地址。
- (2) 指针变量的定义和初始化。
- (3) 间接寻址运算符。
- (4) 按值调用与按地址调用。
- (5) 常用的排序和查找算法。

2.考核要求

- (1) 识记：变量的内存地址。
- (2) 理解：指针变量的定义和初始化、间接寻址运算符。
- (3) 应用：按值调用与按地址调用、常用的排序和查找算法。

二、数据结构部分（40 分）

第一章 绪论

1.考核知识点

- (1) 数据、数据元素、数据项、数据对象、数据结构、逻辑结构、物理结构、元素、结点等基本概念，抽象数据类型的定义、表示和实现方法。
- (2) 算法设计的基本要求以及计算语句频度和估算算法时间复杂度的方法。

2.考核要求

- (1) 识记：有关数据结构的基本概念，四种基本数据结构的特点。
- (2) 理解：四种基本数据结构的基本运算，算法复杂度度量的基本概念。

第二章 线性表

1.考核知识点

- (1) 线性表的定义和基本操作。
- (2) 线性表顺序存储结构的表示和基本运算。
- (3) 线性表链式存储，单链表的表示和查找、插入、删除等基本操作。
- (4) 线性表两种存储结构优缺点的比较。

2.考核要求



- (1) 识记：线性表基本概念、基本运算，单链表的表示。
- (2) 理解：顺序存储和链式存储的比较，各种链表的基本操作算法。

第三章 栈和队列

1. 考核知识点

- (1) 栈和队列的定义及其存储结构、循环队列结构及其存储。
- (2) 栈和队列的主要操作。
- (3) 栈的应用举例，如：数制转换等。

2. 考核要求

- (1) 识记：栈和队列的概念、功能、操作特点、主要运算。
- (2) 理解：栈和队列与一般线性表对比的特殊性，栈和队列的顺序存储和链式存储。
- (3) 应用：栈和队列的常见的使用场合。

第四章 串、数组

1. 考核知识点

- (1) 数组（二维数组）基本概念与存储方式。
- (2) 串的基本概念和基本操作，串的存储方式和 BF 算法。

2. 考核要求

- (1) 识记：数组的顺序存储结构，串的有关概念。
- (2) 理解：二维数组的地址计算，串的基本操作，串的顺序存储结构及其基本操作。

第五章 树和二叉树

1. 考核知识点

- (1) 树的定义和术语。
- (2) 二叉树(完全二叉树、满二叉树)的定义和性质、二叉树的存储结构（顺序表示法和二叉链表表示法）。
- (3) 二叉树遍历算法（先序、中序、后序、层次）。

2. 考核要求

- (1) 识记：树的基本概念。
- (2) 理解：二叉树的存储结构、遍历算法。

第六章 图

1. 考核知识点

- (1) 图的定义，基本术语。
- (2) 图的存储结构，图的遍历，图的连通性与生成树，有向无环图及应用。

2. 考核要求

- (1) 识记：图的基本概念和术语，最小生成树的概念。
- (2) 理解：图的存储方式和基于该存储方式的基本操作（求入度、出度、下一条边等）。



第七章 查找

1.考核知识点

- (1) 查找、关键字、平均查找长度等概念。
- (2) 静态查找表的查找算法(顺序查找、折半查找)。
- (3) 哈希表的特点,构造哈希函数的方法(除留余数法等),哈希表查找和分析。

2.考核要求

- (1) 识记:有关查找的基本概念,静态查找表和动态查找表的概念,哈希表的概念。
- (2) 理解:各种静态查找算法的比较次数分析,哈希函数的选择,冲突处理的方法。

第八章 排序

1.考核知识点

- (1) 排序的目的、分类和排序方法的稳定性的定义。
- (2) 插入排序的思想,直接插入排序的算法。
- (3) 交换排序(冒泡排序的算法,快速排序的思想)。
- (4) 简单的选择排序的算法。

2.考核要求

- (1) 了解:直接插入排序、冒泡排序、简单选择排序算法的特点和适用范围。
- (2) 理解:直接插入排序、冒泡排序、简单选择排序算法、快速排序的思想,上述排序方法的稳定性、平均比较次数、平均移动次数。
- (3) 应用:用类 C 或者 C 语言编写直接插入排序、冒泡排序、简单选择排序等排序算法。

IV.参考书目

- 《C 语言程序设计(第 3 版)》,何钦铭 颜晖主编,高等教育出版社,2015 年 8 月第 3 版。
《数据结构(C 语言第二版)》,严蔚敏、李冬梅编著,人民邮电出版社,2015 年 2 月。

V.题型示例

一、单项选择题(每小题备选答案中,只有一个符合题意的正确答案。请将选定的答案,填入表格中,多选、错选、不选均不得分。)

1.C 语言程序的 3 种基本结构是顺序结构、选择结构和_____结构。

A. 循环 B. 递归 C. 转移 D. 嵌套

2.对于栈操作的原则是_____。

A. 先进先出 B. 后进先出 C. 后进后出 D. 不分顺序

二、填空题
1.当 $a=0$, $b=2$, $c=3$ 时,则表达式 $c+b>0 \ \&\& \ b==a$ 的值是_____。

2.一棵深度为 3(根的层次号为 1)的满二叉树有_____个叶子结点。

三、判断题



1.C 语言中,表达式 $1/2$ 与 $1.0/2$ 的结果相同。()

数据结构中队列是非线性结构。()

四、程序分析题

下列程序的运行结果是_____。 `#include <stdio.h>`

```
int main(void) { int x=8; do {printf("%d", x);} while(!x); return 0;}
```

五、算法设计题

1.求整数的位数及各位数字之和,对于给定的正整数 N,求它的位数及其各位数字之和。

输入示例: 9875 输出示例: 4 29

2.简单选择排序也称作直接选择排序,请简述选择排序算法的基本思想并编写相应程序:输入一个正整数 $n(1 < n \leq 10)$,再输入 n 个整数,将它们从小到大排序后输出。输入示例: Please input n: 5 Please input 5 number: 12 7 5 8 10 输出示例: Select Sort: 5 7 8 10 12

善芽
专升本
18126861610

