

I. 课程性质与课程目标

一、课程的性质和特点

高级语言程序设计是高等教育自学考试计算机科学与技术专业(专升本)考试计划中重要的专业基础课程,为学习其他专业课程奠定程序设计的基础。

本课程的主要目的是通过对 C 语言的语法规则、数据类型、数据运算、语句、数组、函数、指针、结构体类型和文件,以及结构化程序的设计方法和三种基本结构的学习,使学生掌握 C 语言的基础知识和编程方法,为进一步开发各种应用程序打下良好的基础。

二、课程目标

通过本课程的学习,要求掌握 C 语言的基础知识,具备 C 语言编程的基本技能。

1. 基本知识

(1) 熟练掌握 C 语言的各种数据类型(包括基本类型、构造类型、指针类型等)和运算符,能正确运用表达式进行各种数据运算。

(2) 熟练掌握结构化程序设计的三种基本结构(顺序结构、选择结构和循环结构),能运用相关语句实现三种基本结构,实现特定的程序功能。三种基本结构是本课程的学习重点之一。

(3) 熟练掌握 C 语言的一维数组、二维数组和字符数组的定义和使用方法,能运用数组进行程序设计。

(4) 熟练掌握 C 语言自定义函数的定义、一般调用、嵌套调用、递归调用和参数传递方式;了解 C 语言各种库函数的功能和调用方法;能运用函数进行程序设计。函数是本课程的学习重点之一。

(5) 掌握 C 语言指针的定义和使用方法;了解指针数组的定义和使用方法;了解利用指针访问字符串的方法,利用指针访问数组的方法,利用指针作为函数参数进行数据传递的方法;能灵活运用指针进行程序设计。指针是本课程的学习难点之一。

(6) 掌握 C 语言结构体类型的定义和使用方法;了解结构体类型数组的定义和使用方法;了解结构体类型指针的定义和使用方法;了解自定义数据类型的定义和使用方法;能运用结构体类型进行程序设计。

(7) 掌握 C 语言宏定义、文件包含和条件编译三种编译预处理命令的使用方法,能在程序设计中正确应用。

(8) 了解 C 语言文件操作的过程;掌握文件打开和关闭函数,文件读/写函数和文件定位函数的调用方法;能熟练运用文件进行程序设计。文件是本课程的学习难点之一。

(9) 了解结构化程序设计思想。

(10) 了解结构化和模块化程序设计方法。

2. 基本技能

(1) 掌握程序阅读和分析的方法。

(2) 掌握程序设计和调试的方法。

(3) 了解并初步掌握应用程序开发技术。

三、与相关课程的联系

先修课程是计算机应用技术，了解计算机组成原理、软硬件和数制的概念以及上机操作的基本知识。

后续课程包括汇编语言程序设计、数据结构和操作系统等。本课程的学习将为其提供程序设计的基础。

II. 考核目标

四个能力层次从低到高用教育测量学的语言表述依次是：识记、领会、简单应用、综合应用。

识记：要求考生能够对大纲中的知识点有清晰准确的认识，并能做出正确的判断和选择。

领会：要求考生能够对大纲中的知识点有一定的理解，清楚它与其他相关知识点的联系与区别，并能做出正确的表述和解释。

简单应用：要求考生能够运用本大纲中各部分的少数几个知识点，完成简单功能的编程。

综合应用：要求考生在对大纲中的知识点熟悉和理解的基础上，完成复杂功能的编程。

简言之，对四个能力层次的简单通俗的归纳概括可以理解为：“识记”是能够指出“是什么”，“领会”是要回答出“为什么”，“应用”（简单应用、综合应用）是要求回答“做什么”和“怎么做”。

III. 课程内容与考核要求

第一章 概 述

一、学习目的与要求

通过本章学习，了解计算机语言、高级语言、算法、程序和程序设计的概念；掌握常用的算法表示方法以及程序设计方法，为后续章节的学习奠定基础。

二、课程内容

- (1) 计算机发展。
- (2) 计算机语言。
- (3) 算法及其描述方法。
- (4) 程序和程序设计方法。

三、考核知识点和考核要求

1. 计算机语言，达到“识记”层次

- (1) 了解计算机语言的发展。
- (2) 了解高级语言的概念和特点。

2. 程序和程序设计方法，达到“识记”层次

- (1) 了解程序的概念。
- (2) 了解程序设计的方法。

(说明：根据算法绘制算法流程图不作为本课程的考核内容，但是其对于学习 C 程序设计是十分重要的。)

第二章 C 语言基础知识

一、学习目的与要求

通过本章学习，了解 C 语言的字符集、词法和语句分类；了解 C 程序的组成；掌握在 VC (Microsoft Visual C++ 6.0) 环境下，创建源程序、编译、链接、生成可执行程序、编辑修改源程序、运行可执行程序、查看运行结果等基本操作；并通过程序调试，逐步达到熟练掌握的程度。

重点：C 语言的字符集、标识符和关键字；C 语言的基本语句分类；C 程序的基本组成；VC 环境下创建、编辑修改、编译、链接、运行 C 程序的操作；VC 各菜单项的功能。

难点：转义字符；C 程序的组成。

二、课程内容

- (1) C 语言发展和特点。
- (2) C 语言基本词法。
- (3) C 语言基本语句分类。
- (4) C 程序基本组成。
- (5) C 程序开发环境。

三、考核知识点和考核要求

1. C 语言基本词法，达到“识记”层次

- (1) 了解 C 语言的字符集。
- (2) 掌握 C 语言的转义字符。
- (3) 了解 C 语言的标识符构成规则。
- (4) 了解 C 语言的关键字及其含义。

2. C 语言基本语句分类，达到“识记”层次

- (1) 了解 C 语言的语句分类。
- (2) 了解 C 语言各类语句的基本功能。

3. C 程序基本组成，达到“领会”层次

- (1) 理解 C 程序的基本组成。
- (2) 了解书写 C 程序的基本规则。

4. C 程序开发环境，达到“简单应用”层次

- (1) 掌握 VC 启动和退出的操作方法。
- (2) 掌握 VC 环境下创建、编辑修改和保存 C 源程序的操作方法。
- (3) 掌握 VC 环境下编译、链接和生成可执行程序的操作方法。
- (4) 掌握 VC 环境下运行可执行程序 and 查看运行结果的操作方法。

第三章 数据类型、运算符和表达式

一、学习目的与要求

通过本章学习，了解 C 语言的各种数据类型和运算符；掌握各种基本数据类型常量和字

字符串常量的书写方法；掌握符号常量的定义和初始化方法；掌握基本数据类型变量的定义、初始化和赋值方法；掌握有名常量的定义和初始化方法；掌握基本运算符的运算规则、优先级和结合性；能正确构成基本类型的表达式；掌握表达式计算和变量赋值时数据类型转换的规则。

重点：基本数据类型（整型、短整型、长整型、无符号基本整型、无符号短整型、无符号长整型、单精度实型、双精度实型、字符型）常量和字符串常量的书写方法；变量的定义、初始化、赋值以及使用方法；基本运算符（算术、关系、逻辑、赋值、逗号、条件、长度和位运算符）的运算规则、优先级和结合性；表达式（算术、关系、逻辑、赋值、逗号、条件和位运算表达式）的构成规则和计算。

难点：运算符的优先级；复合表达式的计算。

二、课程内容

- (1) 数据类型。
- (2) 常量。
- (3) 变量。
- (4) 运算符和表达式。
- (5) 数据类型转换。

三、考核知识点和考核要求

1. 数据类型的概念及其分类，达到“识记”层次

- (1) 了解数据类型的概念。
- (2) 了解 C 语言中数据类型的分类。

2. 整型、实型、字符型、字符串常量的书写方法，达到“领会”层次

- (1) 掌握长、短整型常量的书写方法。
- (2) 掌握单精度和双精度实型常量的书写方法。
- (3) 掌握字符型常量的书写方法。
- (4) 掌握字符串常量的书写方法。

3. 符号常量的使用，达到“简单应用”层次

- (1) 掌握符号常量的定义方法。
- (2) 能够在程序中正确使用符号常量。

4. 整型、实型、字符型变量的定义和初始化方法，达到“领会”层次

- (1) 掌握整型（整型、短整型、长整型、无符号基本整型、无符号短整型、无符号长整型）变量的定义和初始化方法。
- (2) 掌握实型（单精度实型、双精度实型）变量的定义和初始化方法。
- (3) 掌握字符型变量的定义和初始化方法。

5. 有名常量的定义和初始化方法，达到“领会”层次

- (1) 了解有名常量的特点。
- (2) 掌握有名常量的定义和初始化方法。

6. 基本类型数据在内存的存放方式，达到“识记”层次

- (1) 了解整型（整型、短整型、长整型、无符号基本整型、无符号短整型、无符号长整型）数据在内存的存放方式。

- (2) 了解实型数据在内存的存放方式。
- (3) 了解字符型数据在内存的存放方式。
- (4) 了解字符串常量在内存的存放方式。

7. 基本运算符的运算规则、优先级和结合性，达到“领会”层次

- (1) 掌握算术运算符的运算对象、运算规则、优先级和结合性。
- (2) 掌握关系运算符的运算对象、运算规则、优先级和结合性。
- (3) 掌握逻辑运算符的运算对象、运算规则、优先级和结合性。
- (4) 掌握赋值运算符的运算对象、运算规则、优先级和结合性。
- (5) 掌握逗号运算符的运算对象、运算规则、优先级和结合性。
- (6) 掌握条件运算符的运算对象、运算规则、优先级和结合性。
- (7) 掌握位运算符的运算对象、运算规则、优先级和结合性。
- (8) 掌握长度运算符的运算对象、运算规则、优先级和结合性。

8. 表达式的构成和运算规则，达到“简单应用”层次

- (1) 熟练掌握算术表达式的构成和运算规则。
- (2) 熟练掌握关系表达式的构成和运算规则。
- (3) 熟练掌握逻辑表达式的构成和运算规则。
- (4) 熟练掌握赋值表达式的构成和运算规则。
- (5) 熟练掌握逗号表达式的构成和运算规则。
- (6) 熟练掌握条件表达式的构成和运算规则。
- (7) 熟练掌握位运算表达式的构成和运算规则。

9. 变量赋值和表达式计算时的数据类型转换规则，达到“领会”层次

- (1) 理解不同数据类型的数据和变量在赋值时的“就左不就右”的自动数据类型转换规则。
- (2) 理解不同数据类型的数据在参与表达式计算时的“就高不就低”的自动数据类型转换规则。
- (3) 了解强制数据类型转换方法。

第四章 结构化程序设计

一、学习目的与要求

通过本章学习，了解结构化程序的三种基本结构；熟练掌握赋值语句、函数调用语句、表达式语句、复合语句的格式和功能；熟练掌握字符输入/输出和格式输入/输出函数的调用格式和功能，能正确设计顺序结构程序；熟练掌握单分支、双分支、多分支选择语句的格式和功能，能正确选取选择语句设计选择结构程序；熟练掌握 while 语句、do-while 语句、for 语句、break 语句、continue 语句的格式和功能，并能根据要求选取循环语句设计循环结构程序。

重点：赋值语句、函数调用语句、表达式语句、复合语句的格式和功能；字符输入/输出和格式输入/输出函数的调用格式和功能；if、if-else 和 switch 语句的格式和功能；while 语句、do-while 语句、for 语句、break 语句、continue 语句的格式和功能；顺序结构、选择结构和循环结构的程序设计。

难点：格式输入/输出函数中格式说明符的使用；多分支选择结构的程序设计；二重循

环的程序设计；break 和 continue 语句的正确使用。

二、课程内容

- (1) 结构化程序设计方法。
- (2) 结构化程序三种基本结构。
- (3) 顺序结构程序设计。
- (4) 选择结构程序设计。
- (5) 循环结构程序设计。

三、考核知识点和考核要求

1. 结构化程序的设计方法和三种基本结构，达到“识记”层次

- (1) 了解结构化程序的设计方法。
- (2) 了解顺序结构的特点。
- (3) 了解三种选择结构的特点。
- (4) 了解三种循环结构的特点。

2. 赋值语句、函数调用语句、表达式语句、复合语句的使用，达到“简单应用”层次

- (1) 掌握赋值语句的格式和基本功能。
- (2) 掌握函数调用语句的格式，能正确调用库函数。
- (3) 掌握表达式语句的格式和基本功能。
- (4) 掌握复合语句的格式和基本功能。
- (5) 熟练掌握使用赋值语句、表达式语句解决简单计算问题的方法。

3. 字符输入/输出函数的使用，达到“简单应用”层次

- (1) 熟练掌握字符输入/输出函数的调用方法和功能。
- (2) 能使用字符输入/输出函数解决程序中字符的输入/输出。

4. 格式输入/输出函数的使用，达到“简单应用”层次

- (1) 掌握格式输入/输出函数中格式说明符的含义。
- (2) 熟练掌握格式输入/输出函数的调用方法和功能。
- (3) 能使用格式输入/输出函数解决程序中基本类型数据的输入/输出。

5. 三条选择语句和三种选择结构的程序设计，达到“综合应用”层次

- (1) 能正确使用 if 语句实现单分支选择结构的程序设计。
- (2) 能正确使用 if-else 语句实现双分支选择结构的程序设计。
- (3) 能正确使用 switch 语句实现特定的多分支选择结构的程序设计。
- (4) 能正确使用嵌套的 if-else 语句实现任意的多分支选择结构的程序设计。

6. 三条循环语句和三种循环结构的程序设计，达到“综合应用”层次

- (1) 能正确使用 while 语句实现循环结构的程序设计。
- (2) 能正确使用 do-while 语句实现循环结构的程序设计。
- (3) 能正确使用 for 语句实现循环结构的程序设计。
- (4) 掌握 break 和 continue 语句的功能和使用方法。

7. 利用循环嵌套实现二重循环结构的方法，达到“简单应用”层次

- (1) 掌握二重循环结构的设计方法。
- (2) 能设计二重循环结构的程序。

第五章 数 组

一、学习目的与要求

通过本章学习，掌握一维数组、二维数组、字符数组的定义、初始化和数组元素的引用方法；掌握字符串的存储和处理方法；掌握常用的处理字符、字符串的库函数的使用方法。

重点：一维数组、二维数组、字符数组的定义、初始化和数组元素的引用；字符串存储和处理方法；处理字符串的库函数的使用。

难点：选择和冒泡排序算法及其程序设计；字符串比较算法及其程序设计；二维字符型数组中字符串的处理。

二、课程内容

- (1) 一维数组。
- (2) 二维数组。
- (3) 字符数组和字符串。

三、考核知识点和考核要求

1. 一维数组的定义、初始化和数组元素的引用方法，达到“综合应用”层次

- (1) 熟练掌握一维数组的定义和初始化方法。
- (2) 熟练掌握一维数组元素的下标引用方法。
- (3) 掌握利用次数型单重循环处理一维数组元素的方法。

2. 二维数组的定义、初始化和数组元素的引用方法，达到“简单应用”层次

- (1) 掌握二维数组的定义和初始化方法。
- (2) 掌握二维数组元素的下标引用方法。
- (3) 掌握利用次数型二重循环处理二维数组元素的方法。

3. 字符数组的定义、初始化和数组元素的引用方法，达到“综合应用”层次

- (1) 熟练掌握字符数组的定义和初始化方法。
- (2) 熟练掌握字符数组元素的下标引用方法。

4. 字符数组中存放的字符串的输入/输出和处理方法，达到“简单应用”层次

- (1) 理解单个字符串在一维字符数组中的存放形式。
- (2) 熟练掌握在一维字符数组中存放的字符串的输入/输出方法。
- (3) 掌握多个字符串在二维字符数组中的存放方式。
- (4) 掌握在二维字符数组中存放的字符串的输入/输出方法。

5. 处理字符串的常用库函数及其使用，达到“简单应用”层次

- (1) 能用 `gets()` 函数和 `puts()` 函数实现字符串的输入/输出。
- (2) 掌握字符串连接 `strcat()`、字符串比较 `strcmp()` 和字符串复制 `strcpy()` 函数的使用方法。
- (3) 掌握字符串长度测试 `strlen()` 函数的使用方法。

第六章 函 数

一、学习目的与要求

通过本章学习，了解模块化程序设计方法；熟练掌握自定义函数的定义和调用方法；掌握函数声明的作用；掌握函数调用时的数据传递方式；能正确选取变量的存储类型，了解变

量生存期和作用域的概念，及其在函数设计和调用中的作用；学会设计简单的嵌套调用函数，了解递归函数的概念和一般设计方法；能正确运用书中介绍的各种常用库函数。

重点：函数的定义和调用方法；函数调用时的数据传递方式；常用库函数的使用。

难点：函数调用时参数的值传递和地址传递方式的区分；变量的生存期和作用域，及其在函数设计和调用中的作用；递归函数的设计方法；常用库函数的使用。

二、课程内容

- (1) 函数的概念和模块化程序设计。
- (2) 函数声明。
- (3) 函数的参数和数据传递方式。
- (4) 变量的存储类型和作用域。
- (5) 函数的嵌套调用和递归调用。
- (6) 常用库函数。
- (7) 函数的程序设计实例。

三、考核知识点和考核要求

1. 模块化程序设计方法，达到“识记”层次

- (1) 了解模块化程序设计的方法和特点。 (1)
- (2) 了解 C 语言中如何实现模块化程序设计。 (1)

2. 函数的定义和调用，达到“综合应用”层次 (8)

- (1) 掌握函数的一般结构和定义方法。 2
- (2) 掌握函数的一般调用方法。)
- (3) 能在程序中实现自定义函数的定义和调用。 (2)

3. 函数调用时的数据传递方式，达到“综合应用”层次 (6)

- (1) 掌握函数调用时参数的值传递方式。 6
- (2) 掌握函数调用时参数的地址传递方式。 (1)
- (3) 掌握函数调用时的返回值传递方式。 (2)
- (4) 能利用全局变量在函数间传递数据。

4. 变量的存储类型及其定义方法，达到“领会”层次 (3)

- (1) 掌握变量存储类型的概念。
- (2) 掌握不同存储类型变量的存储分配方式和使用特点。 (1)
- (3) 掌握变量存储类型的定义方法。

5. 变量的生存期和作用域，达到“领会”层次

- (1) 理解局部变量和全局变量的概念和定义方法。
- (2) 了解通过变量的定义可以确定其生存期。
- (3) 了解通过变量的定义可以确定其作用域。

6. 函数的嵌套调用和递归调用，达到“领会”层次

- (1) 掌握函数嵌套调用的方法和特点。
- (2) 掌握递归函数的定义和设计方法。

7. 常用库函数的使用，达到“简单应用”层次

- (1) 熟练掌握常用的数学处理函数的功能和调用方法。

- (2) 熟练掌握常用的类型转换函数的功能和调用方法。
- (3) 熟练掌握常用的字符处理函数的功能和调用方法。
- (4) 熟练掌握其他的常用库函数的功能和调用方法。

第七章 指 针

一、学习目的与要求

通过本章学习，掌握指针和指针变量的概念；能正确定义所需类型的指针变量；能正确地将指针变量指向变量、数组元素、数组或字符串；能正确地利用指针变量引用所指向的变量、一维数组元素、二维数组元素或字符串中的字符；掌握指针型函数的定义和调用方法；了解指针数组的概念和使用方法。

重点：指针变量的定义、初始化和赋值方法；利用指向变量、数组或字符串的指针变量引用变量、数组元素或字符串；获取变量、数组元素和字符串地址的方法。

难点：指针的概念；对指针变量的运算；指针数组元素的使用。

二、课程内容

- (1) 指针和指针变量。
- (2) 指针和数组。
- (3) 指针和字符串。
- (4) 指针和函数。
- (5) 指针数组。
- (6) 指针的程序设计实例。

三、考核知识点和考核要求

1. 地址、指针和指针变量的概念，达到“领会”层次

- (1) 理解地址和指针的关系。
- (2) 了解指针变量的概念和作用。

2. 指针变量的定义、初始化、赋值和一般使用方法，达到“简单应用”层次

- (1) 掌握指针变量的定义和初始化方法。
- (2) 掌握指针变量的赋值和一般使用方法。
- (3) 掌握取地址运算符“&”和指针运算符“*”的使用方法。

3. 指针变量的使用方法，达到“简单应用”层次

- (1) 掌握指向变量的指针变量的使用方法。
- (2) 能使用指向一维数组元素的指针变量引用数组元素。
- (3) 能使用指向二维数组元素的指针变量引用数组元素。
- (4) 掌握指向字符串的指针变量的使用方法。

4. 指针型函数的定义和调用，达到“领会”层次

- (1) 掌握指针型函数的定义方法。
- (2) 掌握指针型函数的调用方法。

5. 指针数组的定义、初始化和引用，达到“领会”层次

- (1) 了解指针数组的定义和初始化方法。
- (2) 掌握指针数组元素的使用方法。

第八章 结构体类型和自定义类型

一、学习目的与要求

通过本章学习，了解结构体类型数据的特点；熟练掌握结构体类型的定义方法，结构体类型变量、数组和指针变量的定义、初始化和结构体类型数据成员的引用方法；掌握自定义类型的定义方法和应用。

重点：结构体类型的特点和定义方法；结构体类型变量、数组、指针变量的定义、初始化和结构体类型数据成员的引用方法。

难点：嵌套的结构体类型数据的处理。

二、课程内容

- (1) 结构体类型定义。
- (2) 结构体类型变量。
- (3) 结构体类型数组。
- (4) 结构体类型指针。
- (5) 结构体类型的程序设计实例。
- (6) 自定义类型。

三、考核知识点和考核要求

1. 结构体类型的定义方法，达到“领会”层次

- (1) 了解结构体类型的特点和应用。
- (2) 掌握结构体类型的定义方法。

2. 结构体类型变量的定义和成员引用方法，达到“简单应用”层次

- (1) 掌握结构体类型变量的定义和初始化方法。
- (2) 掌握结构体类型变量成员的引用方法。
- (3) 掌握结构体类型嵌套的概念和使用。

3. 结构体类型数组的定义和数组元素成员的引用方法，达到“简单应用”层次

- (1) 掌握结构体类型数组的定义和初始化方法。
- (2) 能正确引用结构体类型数组元素的成员。

4. 结构体类型指针变量的定义和使用，达到“简单应用”层次

- (1) 掌握结构体类型指针变量的定义和初始化方法。
- (2) 能利用结构体类型指针变量引用结构体类型数据的成员。
- (3) 能在函数间传递结构体类型数据。

5. 自定义类型的定义和使用，达到“领会”层次

- (1) 了解自定义类型的定义方法。
- (2) 能将基本数据类型定义为自定义类型。
- (3) 能利用自定义类型定义变量、数组和结构体类型等。

第九章 编译预处理

一、学习目的与要求

通过本章学习，掌握宏定义命令及其使用方法；了解文件包含命令的格式和使用方法；

了解条件编译命令的各种格式和功能。

重点：无参宏定义的方法。

难点：有参宏定义的方法。

二、课程内容

(1) 宏定义命令。

(2) 文件包含命令。

(3) 条件编译命令。

三、考核知识点和考核要求

1. 宏定义命令的使用，达到“领会”层次

(1) 掌握宏定义的方法。

(2) 能在程序中正确使用宏名。

2. 文件包含命令的使用，达到“识记”层次

(1) 了解文件包含命令的功能。

(2) 了解文件包含命令的使用方法。

3. 条件编译命令的使用，达到“识记”层次

(1) 了解条件编译命令的功能。

(2) 了解条件编译命令的使用方法。

第十章 文 件

一、学习目的与要求

通过本章学习，掌握缓冲文件系统中有关文件操作的库函数的使用方法，能设计对文件进行简单操作的应用程序。

重点：缓冲文件系统中“文件型指针”的概念；有关文件操作（打开、关闭、读/写、定位）库函数的使用。

难点：有关文件操作的各种库函数的使用，尤其文件读/写函数的种类较多，功能相似，容易混淆。

二、课程内容

(1) 文件概述。

(2) 文件的打开和关闭函数。

(3) 文件的读/写函数。

(4) 文件的定位函数。

(5) 文件的程序设计实例。

三、考核知识点和考核要求

1. 文件的概念，达到“识记”层次

(1) 掌握文件名的组成规则。

(2) 了解文件的分类及其特点。

2. 缓冲文件系统中“文件型指针”的概念，达到“领会”层次

(1) 了解文件型指针的概念和作用。

(2) 掌握文件型指针的定义方法。

(3) 了解标准输入/输出设备的文件型指针。

3. 缓冲文件系统中文件操作的各种库函数的使用, 达到“简单应用”层次

(1) 掌握文件打开和关闭函数的使用方法。

(2) 掌握文件尾测试函数的使用方法。

(3) 掌握文件读/写中字符读/写函数的使用方法。

(4) 掌握文件读/写中字符串读/写函数的使用方法。

(5) 掌握文件读/写中数据块读/写函数的使用方法。

(6) 掌握文件读/写中格式化读/写函数的使用方法。

(7) 掌握文件指针复位函数和文件随机定位函数的使用方法。

附 录

本教材共有三个附录。

附录 A 是 ASCII 码表, 其中的字符 0~9、A~Z、a~z 要求考生记住其对应的二进制代码, 其他字符不要求。

附录 B 是 C 语言运算符及其优先级汇总表, 便于考生学习、复习时查阅。

附录 C 是 C 语言常用库函数汇总表, 便于考生学习、复习时查阅。

实 验 环 节

一、目的和要求

本课程是一门实践性很强的课程, 实验是必不可少的实践环节, 实验的目的是在实践中训练和培养考生实际操作技能, 以及分析问题和解决问题的能力, 加深对 C 语言的理解, 获得程序设计方法和技巧的训练。要求考生掌握 C 程序的设计、运行和调试方法, 熟练掌握 C 程序设计的完整过程。考生可以在学完所有章节后进行集中实验, 也可以在学习相应章节后进行实验, 以加深对已学章节内容的理解, 为后续的学习打下坚实基础。

下面列出五个实验供参考, 主考院校在进行实验考核时可以另行安排考核内容, 但考核的范围和难易程度应大体相当。

二、实验内容

(1) 顺序、选择和循环结构的应用程序设计。

(2) 数组的应用程序设计。

(3) 函数和指针的应用程序设计。

(4) 结构体类型的应用程序设计。

(5) 文件的应用程序设计。

三、实验大纲

1. 实验目的

通过熟悉 C 程序设计、运行和调试的完整过程, 加深对 C 语言的理解, 获得程序设计方法和技巧的训练, 从而使考生能够运用 C 语言进行简单的编程。

2. 实验时数

在单人单机的实验条件下, 总实验时数不得少于 20 学时, 每个实验时数约为 4 学时。

3. 实验内容和要求

实验一 顺序、选择和循环结构的应用程序设计

内容：编写两个简单的顺序结构程序、三个简单的选择结构程序和三个简单的循环结构程序。

要求：熟悉输入/输出函数、赋值语句、复合语句、三种选择语句、三种循环语句的使用；掌握创建、编辑、编译、链接和运行 C 程序的操作方法和过程。

实验二 数组的应用程序设计

内容：编写一个含有一维数组和单重循环结构的程序；编写一个含有二维数组和二重循环结构的程序。

要求：熟悉数组的定义、初始化、数组元素引用和三种循环语句的使用。

实验三 函数和指针的应用程序设计

内容：编写一个利用指针处理一维数组的程序；编写一个含有一个主函数和两个自定义函数的程序，其中一个函数使用值传递方式传递数据，另一个函数使用地址传递方式传递数据。

要求：熟悉指针变量的定义、初始化和引用；熟悉函数的定义和调用；掌握函数之间值和地址传递方式。

实验四 结构体类型的应用程序设计

内容：编写一个含有结构体类型变量的程序，包括结构体类型变量的输入、处理和输出；编写一个含有结构体类型数组的程序，包括结构体类型数组的输入、处理和输出。

要求：熟悉结构体类型的定义、结构体类型数据的定义、初始化和成员引用方法。

实验五 文件的应用程序设计

内容：编写两个对文件进行创建和操作的程序，包括对文件的读/写。

要求：熟悉文件型指针的定义和使用，以及文件操作函数的使用。

4. 说明

上述实验内容中的程序设计题目可以从本教材的各章习题中选取，每个实验中程序的数目可根据实验时间确定。

IV. 关于大纲的说明与考核实施要求

一、自学考试大纲的目的和作用

课程自学考试大纲是根据专业自学考试计划的要求，结合自学考试的特点而确定的。其目的是对个人自学、社会助学和课程考试命题进行指导和规定。

课程自学考试大纲明确了课程学习的内容以及深度广度，规定了课程自学考试的范围和标准。因此，它是编写自学考试教材和辅导书的依据，是社会助学组织进行自学辅导的依据，是自学者学习教材、掌握课程内容知识范围和程度的依据，也是进行自学考试命题的依据。

二、自学考试大纲与教材的关系

课程自学考试大纲是进行学习和考核的依据，教材是学习掌握课程知识的基本内容与范围的体现，教材内容是大纲所规定的课程内容的详解和扩展。课程内容在教材中可以体现一定的深度或难度，但在大纲中对考核的要求要适当。大纲与教材所体现的课程内容应基本一

致：大纲里面的课程内容和考核知识点，教材里要有。反过来教材里有的内容，大纲里可不一定体现。

三、关于自学教材

《高级语言程序设计》，全国高等教育自学考试指导委员会组编，郑岩编著，机械工业出版社出版，2017 年版。

四、关于自学要求和自学方法的指导

本大纲的课程基本要求是依据专业考试计划和专业培养目标而确定的。课程基本要求还明确了课程的基本内容，以及对基本内容掌握的程度。基本要求中的知识点构成了课程内容的主体部分。因此，课程基本内容掌握程度、课程考核知识点是高等教育自学考试考核的主要内容。

为了有效地指导个人自学和社会助学，本大纲已按章详细给出知识点和具体的考核要求，课程的重点和难点，在章节的基本要求中一般也指明了章节内容的重点和难点。

本课程共 4 学分，其中包含实验考核的 1 学分。

为了帮助大家提高自学效果，给出以下几点建议。

1. 在整体上把握课程的主线

本课程的主要目的是掌握一门高级程序设计语言（相当于学习一种文字），编写在计算机上运行的程序（相当于撰写一篇文章）。因此，课程的内容是围绕着编写程序展开的，学习的主线是：字符集（相当于字母）、词类（相当于单词）、语句（相当于句子）和程序（相当于文章）。

2. 在细节上把握关键的知识点

（1）数据是程序加工处理的对象，反映数据特点的数据类型是关键。C 语言的数据类型比较复杂，主要分为基本类型、构造类型、指针类型和空类型等。掌握数据类型是学习的重点，某些数据类型还是学习的难点。

（2）C 语言的运算符十分丰富。学习时可对运算符进行分类，认真理解，经常使用以加深记忆。由多种运算符组成的复合表达式的计算将取决于运算符的优先级和结合性。

（3）顺序结构比较简单，即按照语句的顺序依次执行。

（4）选择结构分为三种：单分支、双分支和多分支。分别对应三条不同的选择语句。选择结构程序设计的关键在于构造合适的选择条件和选取合适的选择语句。

（5）循环结构也分为三种：当型、直到型和次数型。分别对应三条不同的循环语句。循环结构程序设计的关键在于选取循环结束的条件和构造循环的内容（循环体），有时还需要注意进入循环体前某些变量的初值设置。数组的操作通常和循环有关，通过循环逐一处理数组元素。

（6）C 程序是典型的模块化结构，由一个主函数和若干个函数组成。每个函数相当于一个模块。C 程序总是从主函数开始执行，并在主函数中结束，执行过程中可能调用其他函数。用户自定义函数时，关键是确定主调函数和被调函数之间即形参和实参之间数据传递的方式。

（7）指针变量是用来存放地址的变量，“*指针变量”是其存放的地址对应的变量。

（8）结构体类型是若干个数据类型相同或不同的数据的结合。需要掌握结构体类型的定义方法，结构体类型变量、数组、指针变量的定义方法，结构体类型成员的引用方法。

(9) 文件是用于保存数据的，文件型指针是文件操作的关键，文件的操作主要包括打开、关闭和读/写等，需要调用相应的库函数实现，关键是掌握这些库函数的调用方法。

3. 紧跟大纲学习和备考

考生在学习、复习和备考时，都应仔细阅读本大纲中各章列出的考核知识点和考核要求。每章学习完毕，应按照考核知识点和考核要求进行总结和自我考查。

4. 充分利用每章的习题

独立完成每章的习题是复习、检查和自我考核的有效方法。

5. 充分利用上机实验

在计算机上编程，以提高分析和设计程序的实践能力。

五、对社会助学的要求

(1) 助学单位和老师应熟知本大纲的各项要求和规定。

(2) 教学过程中应以本大纲为依据，以本大纲规定的教材进行教学和辅导，不应增删内容或更改要求。

(3) 助学辅导时应重视基础知识的学习和应用技能的培养，根据考生的特点，按照本大纲的具体要求制订并实施教学计划。

(4) 应注重培养考生的自学能力，使考生能自己提出问题、分析问题和解决问题。

(5) 应使考生了解试题的难易程度和能力层次的高低要求不完全等同，每个能力层次中都可能出现不同难度的试题。

(6) 助学单位应具备上机的条件和环境。

六、对考核内容和考核目标的说明

本课程要求考生学习和掌握的知识点都是考核的内容。课程中各章的内容均由若干知识点组成，在自学考试称为考核知识点。因此，课程自学考试大纲中所规定的考试内容是以分解为考核知识点的方式给出的。由于各知识点在课程中的地位、作用以及知识自身的特点不同，自学考试将对各知识点分别按四个认知（或叫能力）层次确定其考核要求。

七、关于考试命题的若干规定

(1) 考试方式。理论考试为闭卷、笔试。考试时间为 150 分钟。评分采用 100 分制，60 分为及格。考试时，只允许携带笔和橡皮，答卷规定用黑色或蓝色水笔。

实验考试为开卷、上机。考试时间为 90 分钟。评卷采用四级分制：优秀、良好、合格、不合格。合格及以上分数为及格。考试时，只允许携带水笔和铅笔。主考院校应配备必要的草稿纸。

(2) 本大纲各章所规定的基本要求和知识点都属于考核内容。考试命题既要覆盖到章，又要避免面面俱到。要注意突出课程的重点、章节的重点，加大重点内容的覆盖度。

(3) 命题不应有超出大纲中考核知识点范围的题目，考核目标不得高于大纲中所规定的相应的最高能力层次要求。命题应着重考核自学者对基本知识是否了解或掌握，对基本方法是否会应用或熟练应用。不应出与基本要求不符的偏题或怪题。

(4) 本课程在试卷中对不同能力层次要求的分数比例大致为：识记占 20%，领会占 30%，简单应用占 30%，综合应用占 20%。

(5) 要合理安排试题的难易程度，试题的难度可分为：易、较易、较难和难四个等级。每份试卷中不同难度试题的分数比例一般为：2:3:3:2。

必须注意试题的难易程度与能力层次有一定的联系，但二者不是等同的概念。在各个能力层次中都可以有不同的难度。在大纲中要特别强调这个问题，应告诫考生切勿混淆。

(6) 课程考试命题的主要题型包括单项选择题、填空题、程序分析题、程序填充题和程序设计题。

在命题工作中必须按照本课程大纲中所规定的题型命制，考试试卷使用的题型可以略少，但不能超出本课程对题型的规定。

V. 题型举例

一、单项选择题（在每小题后的四个备选项中只有一个是符合题目要求的，请将其选出并将代码填写在题后的括号内）

1. 字符'A'的 ASCII 码值是 65，设 `char c1='A', c2='D';`，则执行语句 `printf(" %d, %d\n", c1, c2-2);`后的输出结果是【 】。

- A. A, D B. A, B C. 65, 68 D. 65, 66

2. 设 `int a=3,b=4;`，则执行语句 `printf("%d,%d",(a,b),(b,a));`后的输出结果是【 】。

- A. 3,4 B. 4,3 C. 3,4,4,3 D. 4,3,3,4

二、填空题（请在每小题的横线处填写正确内容）

1. 能正确表述“ $2 < x < 3$ 或 $x < -10$ ”的 C 语言表达式是_____。

2. 若函数的形参和实参都是数组，则函数之间的数据传递方式是_____。

三、程序分析题

1. 阅读下列程序，写出程序运行的输出结果。

```
#include <stdio.h>
int main(void)
{
    int x=1,a=0,b=0;
    switch(x)
    {
        case 0: a++; break;
        case 1: b++;
        case 2: a++; b++; break;
        case 3: a++; b++;
    }
    printf("a=%d, b=%d\n",a,b);
    return 0;
}
```

2. 阅读下列程序，写出程序运行的输出结果。

```
#include <stdio.h>
int funs(int n)
{
    if(n==1||n==2) return 2;
```

```

        else return(n+funcs(n-1));
    }
    int main(void)
    {
        int x=4;
        printf("S=%d\n", funcs(x));
        return 0;
    }

```

四、程序填空题（请将下列程序横线处缺少的部分补上，使其能正确运行）

1. 从键盘输入 10 个整数，用“冒泡排序法”由小到大进行排序。

```

#include <stdio.h>
#define N 10
int main(void)
{
    int i,j,t,a[N];
    for(i=0;i<N;i++) scanf("%d",&a[i]);
    for(i=0;i<N-1;i++)
        for(j=N-1;j>=i+1; _____) /*第一空*/
            if(a[j]<a[j-1])
                { t=a[j]; _____; a[j-1]=t;} /*第二空*/
    for(i=0;i<N;i++) printf("%5d", _____); /*第三空*/
    printf("\n");
    return 0;
}

```

2. 统计文本文件 file.txt 的行数（提示：以'\n'作为每行的结束）。

```

#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    int lines= _____; /*第一空*/
    char ch;
    FILE *fp;
    fp=fopen("file.txt", "r");
    while( _____) /*第二空*/
    {
        ch=fgetc(fp);
        if(ch _____ '\n') lines++; /*第三空*/
    }
    printf("Lines=%d\n",lines);
    fclose(fp);
    return 0;
}

```

五、程序设计题

1. 从键盘输入实数 x ，按下列公式计算并输出 x 和 y 的值（小数点后保留两位）。

$$y = \begin{cases} \sin x & x \leq 0 \\ x^2 + 1 & 0 < x \leq 10 \\ \frac{1}{x^3 + x^2 + 1} & x > 10 \end{cases}$$

2. 编写程序实现下列功能。

(1) 编写一个摄氏 (C) 到华氏 (F) 温度转换的函数，函数首部是 `float temp(float c)`。

计算公式： $F = \frac{9}{5}C + 32$

(2) 编写主函数从键盘输入摄氏温度的值，并调用上述函数输出摄氏及相应华氏温度的值（小数点后保留两位）。