



《程序设计基础》 教学大纲

2017 年 3 月 9 日

目 录

第一部分 大纲说明	3
1.1 制定依据	3
1.2 适用范围	3
1.3 课程性质	3
1.4 教学目标	4
1.5 课程背景	4
第二部分 教学设计	5
2.1 教学手段	5
2.2 授课思路	5
2.3 学时分配	6
2.4 课程考核	7
第三部分 目标细化	7
3.1 引言	7
3.1.1 主要内容	7
3.1.2 教学目标	7
3.1.3 重点难点	8
3.2 构成C语言程序的单词	8
3.2.1 主要内容	8
3.2.2 教学目标	8
3.2.3 重点难点	9
3.3 从问题求解到程序设计	9
3.3.1 主要内容	9
3.3.2 教学目标	10
3.3.3 重点难点	10
3.4 运算符与表达式	10
3.4.1 主要内容	10
3.4.2 教学目标	11
3.4.3 重点难点	11
3.5 控制流与面向过程的程序设计	11
3.5.1 主要内容	11
3.5.2 教学目标	11
3.5.3 重点难点	12
3.6 指针变量	12
3.6.1 主要内容	12
3.6.2 教学目标	12
3.6.3 重点难点	13
3.7 数组	13
3.7.1 主要内容	13
3.7.2 教学目标	13

3.7.3 重点难点	14
3.8 函数.....	14
3.8.1 主要内容	14
3.8.2 教学目标	14
3.8.3 重点难点	15
3.9 预处理.....	15
3.9.1 主要内容	15
3.9.2 教学目标	15
3.9.3 重点难点	15
3.10 自定义数据类型	16
3.10.1 主要内容	16
3.10.2 教学目标	16
3.10.3 重点难点	16
3.11 标准库函数	16
3.11.1 主要内容	16
3.11.2 教学目标	17
3.11.3 重点难点	17
3.12 项目实战	17
3.12.1 主要内容	17
3.12.2 教学目标	17
3.12.3 重点难点	17
第五部分 教学相关资料.....	18
教材.....	18
教学参考书及网站.....	18

第一部分 大纲说明

1.1 制定依据

本教学大纲是依据河北师范大学软件学院软件工程专业教学计划而制定。

1.2 适用范围

本大纲适用于河北师范大学软件学院软件工程专业的本科生教学。

1.3 课程性质

对高级语言和程序设计的学习和训练被认为是计算机专业的基本功之一。

《程序设计基础》是计算机科学与技术专业的一门重要的专业基础课，也是软件工程专业的一门院专业必修课。课程主要包括两部分内容：C 语言相关知识和程序设计技术。C 是一种高效而又实用的面向过程的程序设计语言，是编程人员广泛使用的编程工具之一。C 语言相关知识包括：C 语言语法和标准库（标准库实际也被认为是 C 语言标准的一部分）。程序设计技术包括：可移植性、性能、编码规范、结构化程序设计、测试和调试等。

本课程在大学一年级的第二学期开设，是软件学院软件工程专业学生接触的第一种程序设计语言，意义重大。具体表现在：1) 第一门语言对于培养学生的编程兴趣十分重要。2) 作为面向对象程序设计语言 C++ 的前驱课程，本课程对于 C++ 的学习非常重要。原因在于：C++ 是在 C 语言基础上“扩展”得到的一门语言。要想学好 C++，C 语言的基础必须十分扎实。3) 本课程对于数据结构课程的学习同样十分重要。目前，数据结构教材普遍采用 C 语言版的。如果对于 C 语言中的主要知识点，如：结构体、指针、递归函数等没有掌握好，理解数据结构中的 ADT 及相关算法是有一定困难的。

通过课程的学习，要求学生扎实掌握 C 语言语法，能够对常规问题熟练编写程序，并对各种程序设计技术有一定程度的认识。通过课程的学习为今后学习数据结构和面向对象程序设计等课程打下坚实的基础。

1.4 教学目标

通过本课程的学习，要求学生达到下列基本目标：

知识要求	<ul style="list-style-type: none">(1) 了解C语言的发展历史，懂得“标准”的来源及意义。(2) 掌握标准与实现的关系。掌握C语言与程序设计之间的关系。(3) 知道C语言的基础知识，包括数据类型、表达式与运算符、声明、语句、函数（尤其是递归函数的分析）、数组、自定义数据类型、指针、宏、标准库等。(4) 初步掌握结构化程序设计的基本知识，如：算法分析与设计、流程图。(5) 重点掌握：语句（尤其是循环语句）、递归函数的设计与分析、结构体、指针。使学生在《数据结构》课程时，不存在C语言的障碍。(6) 初步了解C语言与C++的关系与区别。阐述C语言中抽象、封装、多态、异常处理的概念，为学生学习C++打下基础。(7) 了解C语言的可移植性。(8) 了解大型程序设计和组织的思想和方法。
能力要求	<ul style="list-style-type: none">(1) 熟练运用Microsoft VS2012/Dev-C++ 编辑、编译、调试程序。(2) 能够对英语的调试信息进行识别并进行有效调试。(3) 具备阅读、分析中小型C语言项目代码的能力。(4) 熟练运用C语言的实现一些常见的算法。(5) 熟练运用C语言的相关知识处理简单的实际问题。(6) 能够运用C语言完成一个中等程度的实际项目。(7) 掌握从分析问题到编写伪代码、编写程序、调试程序和测试程序的程序设计思路。(8) 掌握高质量编程的一些常用规则和编程规范。
素质要求	<ul style="list-style-type: none">(1) 热爱自己的祖国。(2) 良好的编程习惯。(3) 吃苦耐劳的精神。(4) 强烈的团队意识。(6) 良好的法制与诚信意识。(7) 具有吸收新技术和知识的能力。(8) 具有创新意识。

1.5 课程背景

1、学习本课程之前需要学习《计算机导论》。

2、本课程的后续课程是《面向对象程序设计 C++》、《数据结构》。

第二部分 教学设计

2.1 教学手段

教学手段分两种：理论教学和实践教学。理论教学在课上完成，采用多媒体教学的手段，主要借助短小精悍的示例代码来介绍重要的概念、重要的思想和重要的方法。实践教学也在课上进行，完成不了的利用课下时间，形式分两种：实验教学和课程设计。实验教学要求学生根据实验手册中的实验要求，完成相应的编码、调试和测试，对理论教学中的方法和思想进行模仿和复现，达到强化编程技能，强化对重要概念、重要思想和重要方法的理解和掌握的目的。课程设计则通过一个相对完整的开发需求，把课程中涉及的重点知识和技巧进行综合的运用。课程设计在期中之后进行，通过一个相对完整的开发需求，把课程中涉及的大部分语法知识和程序设计知识进行综合运用，总代码行数应不少于 500 行。

课后程序作业要求学生根据《C 程序设计习题集》中题目的具体要求，独立完成程序的编码、调试和测试，对理论教学中的方法和思想进行模仿和复现，达到强化编程技能，强化对重要概念、重要思想和重要方法的理解和掌握的目的。

2.2 授课思路

就知识点而言，比较难以掌握的知识有：循环语句、数组、结构体、指针、递归函数。这些难点所涉及的知识涵盖了 C 语言的所有知识点。比如：“语句”涉及了变量的声明、表达式与运算符、关键字等；“函数”涉及了声明符、数据类型、语句等。因此，本课程的授课思想是：1) 主要知识点围绕着内存展开阐述。2) 主要围绕关键知识（即重点与难点知识）展开讲述。具体而言：

- 数据类型：数据在内存中的存储方式、数据类型转换的实质及转换带来的隐患。
- 循环语句：展示循环语句的执行过程的详细分析，让学生深入理解循环语句上初始表达式、循环表达式、条件表达式的作用及循环语句的功能。

- 内存的五大区域及作用：堆区、栈区、文字常量区、全局区、代码区。
- 指针：指针及所指向的变量在内存中的关系示意图。二重指针的内存存储示意图。const 指针的内存存储示意图。
- 数据对象（内置类型、自定义类型）：对象在内存中是如何存储的。变量名、内存地址、数据对象之间的关系。
- 数组：讲述数组这一“容器”中的元素在内存中的存储。讲述指针与数组的关系。
- 递归函数：结合递归函数的运行分析，让学生体会递归函数的执行流程及压栈、出栈的过程。
- 从内存的角度阐述参数传递的本质。
- 封装：结构体与封装的概念。
- 多态：函数指针与静态多态的概念。

在上述知识的基础上，再展开其他知识的讲授。

具体授课的方式：提出问题→解决问题→知识归纳→问题深化（拓展思考）。

2.3 学时分配

本课程总学时为 81 学时，共 3.5 学分。其中理论教学 45 学时，实践教学 36 学时。这两部分教学课时分配情况分别如下。

注：1) 实践教学中的课程设计分配课时，即不占用上课时间。2) 若实践教学中的实验教学课上不能完成，就作为课下的作业，继续完成。

序号	主要内容	学时分配	理论课时	实验课时
1	引言	5	3	2
2	构成C语言程序的单词	5	3	2
3	从问题求解到程序设计	5	3	2
4	运算符与表达式	5	3	2
5	控制流与面向过程的程序设计	10	6	4
6	指针	10	7	3
7	数组	9	4	5
8	函数	10	6	4
9	预处理	5	3	2
10	自定义数据类型	5	3	2

11	标准库函数	5	2	3
12	项目实战	7	2	5
合计		81	45	36

注：

1. 上表不代表实际的教学进度安排，只反映课时的总体分布情况；章节的顺序也不代表实际课程的讲解顺序，实际讲授次序根据学生实际情况会有所调整。
2. 习题课可由任课教师根据实际教学情况灵活安排。

2.4 课程考核

期末考核采用期末笔试加上机考核方式。笔试为闭卷。

本课程的成绩由三部分组成：

- 1、期中考试成绩：30 分
- 2、平时成绩：40 分（作业、出勤、纪律、综合表现等）
- 2、期末考试成绩：30 分

第三部分 目标细化

3.1 引言

3.1.1 主要内容

1. 语言与程序的关系。
2. C 语言与标准演化史。
3. 标准与实现的关系。
4. C 语言程序的编辑、编译、连接、运行程序的体验。

3.1.2 教学要求

1. 初步理解“程序”的概念。
2. 掌握标准与实现的关系。

3. 掌握编辑 C 程序的方法，理解源字符集。
4. 初步理解翻译的步骤，掌握“编译”、“连接”和“运行”程序的方法。
5. 学会 Visual Studio 2012 的安装。
6. 能模仿例题编辑、编译、连接和运行另外一个程序。

3.1.3 重点难点

1. 语言与程序的关系。
2. 标准与实现的关系。
3. 程序的编辑、编译、链接的过程。

3.2 构成 C 语言程序的单词

3.2.1 主要内容

1. 详解第一个程序。
2. 程序与“单词”。
3. 一种特殊的“单词”：关键词。
4. 让学生体会到本课程要学习什么样的内容：C 语言的语法、程序设计的知识及方法。
5. 源文件、头文件、编译单元的概念。
6. 注释的作用。
7. main 函数的作用及常见形式。
8. 在 VS2012 IDE 环境中新建项目、添加源文件/头文件、编辑、编译、连接、运行程序的过程。

3.2.2 教学要求

1. 了解构成 C 语言程序的单词类别。
2. 了解 C 语言关键词。
3. 初步理解头文件的作用。

4. 初步理解“连接”的概念。
5. 掌握注释的使用。
6. 掌握 `main` 函数的作用及推荐的写法。
7. 掌握使用 Visual Studio2012（简称为 VS2012）建立 C 语言项目，添加/编辑源文件、头文件的方法，掌握在 VS2012 中“编译”、“连接”、“运行”程序的方法。

3.2.3 重点难点

1. 本课程知识总揽：C 语言（单词及单词的含义、用法）、程序设计。
2. `main` 函数的作用。
3. 从源程序到可执行文件的步骤。

3.3 从问题求解到程序设计

3.3.1 主要内容

1. 问题求解与算法：程序为什么能解题？程序必须解题步骤才能解题，解题步骤与算法。
2. 程序与算法的关系。
3. 对同一个题，有多种算法，算法的好坏如何评价？
4. 算法的五大特征。算法的本质是处理数据。
5. 数据与数据类型。
6. 数据类型与标识符，标识符的声明。
7. 常量与字面值。字面值的数据类型。字符串字面值的特殊性：自动加上一个结束符。
8. 数据的输入输出。算法的特征：至少一个输出、零个输入。显然讨论输入、输出是有意义的。

3.3.2 教学要求

1. 了解算法与程序的关系。
2. 掌握 C 语言有哪些数据类型，不同的数据类型有什么区别？
3. 掌握变量的声明方法。
4. 掌握合法标识符。
5. 理解常量与字面值的区别。
6. 掌握不同类型的字面值的写法。
7. 掌握数据的输出与输入方式，重点掌握“转换说明”。

3.3.3 重点难点

1. 算法与程序的关系。算法、程序、C 语言三者之间的关系。
2. 数据类型、标识符、数据对象、内存地址之间的关系。
3. 转换说明。

3.4 运算符与表达式

3.4.1 主要内容

1. 左值与右值。
2. 运算符总揽及运算符优先级。
3. 运算符与表达式：算法运算符及表达式、赋值运算符与表达式、关系/判等运算符与表达式、逻辑运算符与表达式、条件运算符与表达式、位操作运算符与表达式、逗号运算符与表达式、sizeof 运算符。
4. 表达式副作用、括号运算符与表达式。
5. 类型转换：变量到变量、字面值到变量、表达式中的类型转换分析。
6. 溢出简介。

3.4.2 教学要求

1. 理解左值及右值。
2. 掌握运算符的种类、重点掌握运算符优先级。
3. 熟悉各种运算符的功能及相关表达式的求值方法。
4. 了解 `sizeof` 运算符。
5. 了解表达式副作用。
6. 掌握显式类型转换的方法，了解隐式转换。
7. 掌握溢出的计算方法，了解在什么情况下可能会造成溢出。

3.4.3 重点难点

1. 运算符优先级。
2. 表达式求值分析。
3. 类型转换。

3.5 控制流与面向过程的程序设计

3.5.1 主要内容

1. 算法及基本结构：顺序结构、选择结构、循环结构。
2. 算法流程图的画法。
3. 语句的定义与分类。
4. 算法基本结构与 C 语言三类语句：顺序语句、选择语句、循环语句。
5. 面向过程的程序设计简介。

3.5.2 教学要求

1. 掌握算法的三种基本结构。
2. 初步了解面向过程的程序设计。
3. 理解控制算法运行路径的方法。

4. 掌握逻辑运算符、逻辑表达式在选择结构、循环结构中的作用。
5. 熟练使用顺序结构、条件结构、循环结构，学会算法的伪代码/N-S 结构的描述法。
6. 掌握语句的概念，熟练使用 if-else、switch、for、while、do、goto 等语句。
7. 掌握多重循环的执行分析方法。
8. 能设计简单的算法，并根据算法编写程序。
9. 能读懂算法流程，并根据算法编写程序。

3.5.3 重点难点

1. 流程图。
2. 循环结构与循环语句。
3. 面向过程的程序设计。

3.6 指针变量

3.6.1 主要内容

1. 单重、多重指针变量的声明。
2. 指针变量本身的存储单元与指针变量所指的存储单元。
3. 阐述变量名、数据对象、存储地址、存储空间的大小、指针变量之间的关系。
4. 特殊指针：const 指针、空指针、void 指针。
5. 指针变量的运算。

3.6.2 教学要求

1. 掌握指针声明符、指针变量的声明方法。
2. 理解指针变量的两个关键点：存放地址、“捆绑”一块内存空间。
3. 理解单重及多重指针的赋值，掌握通过指针访问所指内存空间中数据对象

的方法。

4. 理解 `const` 指针。
5. 了解空指针及通用指针的作用。
6. 了解指针变量的运算。

3.6.3 重点难点

1. 指针的两大属性：类型、存放地址。类型决定了指针所指向单元的大小。
2. `const` 指针。
3. 指针运算。

3.7 数组

3.7.1 主要内容

1. 数组的声明。
2. 一维数组的存储、初始化方法及使用。
3. 二维数组的存储、初始化方法及使用。
4. 一维字符数组与字符串（重点回顾字符串字面值的特殊性：自动在背后加上一个结束符）。
5. 指向数组的指针与元素为指针的数组。重点阐述：数组存储空间与指针类型之间的关系。
6. 指针的运算（针对指向数组的指针）。

3.7.2 教学要求

1. 掌握数组声明符、数组声明的方法。
2. 掌握一维数组、二维数组在内存中的存储。
3. 掌握通过下标方式访问数组中各元素的方法。
4. 掌握通过指向数组的指针访问数组中各元素的方法。
5. 掌握字符数组与其他内置类型数组之间的细微差别。

6. 重点掌握指向数组的指针及指针数组的声明、使用。

3.7.3 重点难点

1. 指向数组的指针与元素为指针的数组。
2. 针对指向数组的指针的运算。

3.8 函数

3.8.1 主要内容

1. 面向过程的程序设计与函数的关系。
2. 函数的概念。
3. 函数的定义、声明。函数原型的概念。
4. 参数传递：值传递与地址传递。重点：如何将数组传递给指针？
5. 嵌套调用与递归调用。递归函数的执行分析。
6. 生存期与声明作用域。
7. 存储类说明符。
8. 类型限制符。
9. 动态内存分配标准库函数。
10. 指向函数的指针。函数指针与多态。
11. typedef 的高级用法。

3.8.2 教学要求

1. 理解“函数”与“面向过程的编程”的关系。
2. 掌握函数原型声明与原型定义的方法。
3. 深入理解参数传递，尤其是“传递地址”的情况。
4. 理解生存期与作用域的概念。
5. 了解函数指针的声明、使用。
6. 掌握 typedef 的各种用法。

7. 了解程序在内存中的存储，掌握如何从堆中申请内存，如何使用和释放堆中内存。
8. 掌握嵌套调用、递归函数的分析方法。

3.8.3 重点难点

1. 参数传递。
2. 递归调用。
3. 生存期与声明作用域。

3.9 预处理

3.9.1 主要内容

1. 预处理器与预处理指令。
2. 文件包含：`#include`与`#include <>`的区别。
3. 宏（对象式、函数式、预定义）定义与宏替换。
4. 条件编译。

3.9.2 教学要求

1. 理解预处理指令的作用及给程序员带来的好处。
2. 重点掌握`#include`、`#define`预处理指令。
3. 掌握宏定义的方法，学会分析宏替换的详细过程。
4. 掌握条件编译指令的作用及给程序移植、调试等带来的好处。
5. 了解预定义宏，学会使用预定义宏。

3.9.3 重点难点

1. 条件编译及其作用。

3.10 自定义数据类型

3.10.1 主要内容

1. 枚举类型及使用。
2. 结构体类型的声明。
3. 结构体对象、结构体对象数组。
4. 结构体对象成员的访问。
5. 结构体与动态链表。
6. 共同体。

3.10.2 教学要求

1. 了解为什么需要自定义数据类型。
2. 掌握三种自定义数据类型的定义语法。
3. 掌握结构体对象、共同体对象在内存中的存储。
4. 掌握在栈区和堆区创建结构体对象的方法。
5. 掌握结构体对象指针、结构体对象数组的定义、使用方法。
6. 掌握枚举类型变量的使用。
7. 初步了解数据结构的相关知识。

3.10.3 重点难点

1. 结构体、结构体数据。
2. typedef 与结构体。

3.11 标准库函数

3.11.1 主要内容

1. 标准库与非标准库。

2. 字符与字符串处理函数。
3. 文件及相关函数。
4. 标准语言补充。

3.11.2 教学要求

1. 掌握标准库的作用。
2. 了解有哪些标准头文件，各头文件中主要声明什么样的函数。
3. 掌握字符及字符串处理的标准库函数的用法。
4. 掌握内存管理方面的标准库函数的用法。
5. 掌握文件操作的标准库函数，掌握文件与内存间数据“交换”的方法。
6. 掌握标准输入输出函数的使用方法及区别。
7. 了解标准语言补充方面的相关知识。

3.11.3 重点难点

1. 文件及文件操作。

3.12 项目实战

3.12.1 主要内容

1. 对一个项目，阐述从需要到设计到分析到编码的全过程。

3.12.2 教学要求

1. 利用 C 语言完成一个中小型项目。
2. 对课程内容有个整体的把握和更深入的理解。

3.12.3 重点难点

1. 项目需求分析、设计。

第四部分 教学相关资料

教材

- [1] 李文斌等编著《C 语言与程序设计大学教程》，清华大学出版社。
- [2] 李文斌等编著《C 语言与程序设计大学教程习题与实验手册》，清华大学出版社。

教学参考书及网站

- [1] 谭浩强著. C 程序设计. 北京: 清华大学出版社, 2005
- [2] Brian W. Kernighan, Dennis M. Ritchie 著; 徐宝文, 李志译, C 程序设计语言 (第 2 版·新版), 机械工业出版社, 2001
- [3] H.M.Deitel, P.J.Deitel 著, 薛万鹏等译, C 程序设计教程, 机械工业出版社, 2000
- [4] Stephen Prata 著, 云巅工作室译, C Primer Plus (第五版), 人民邮电出版社, 2005
- [5] Al Kelley, Ira Pohl, A Book on C: Programming in C, 4th Edition, 机械工业出版社, 2004
- [6] 苏小红, 陈惠鹏, 孙志岗, C 语言大学实用教程, 电子工业出版社, 2004
- [7] ISO/IEC 9899:1999 (C99)
- [8] Perter Van Der LinDen 著, 徐波译, C 专家编程, 人民邮电出版社, 2002
- [9] 林锐, 韩永泉, 高质量程序设计指南——C++/C 语言 (第二版), 电子工业出版社, 2003
- [10] Brian W. Kernigham 著, 裘宗燕译, 程序设计实践, 机械工业出版社, 2003
- [11] Samuel P. Harbison; Guy L. Steele 著, 徐波译, C 语言参考手册, 机械工业出版社, 2008

执 笔 人：丁盟

审 定 人：

批 准 人：