河北师范大学《程序设计基础》课程教学大纲

（理论课程）

课程代码：32201108

课程名称：程序设计基础

英文名称：Foundations Of Programming

授课语言：汉语

开课单位：软件学院

大纲制定人：丁盟

大纲审定人：丁盟

一、课程说明

**1.课程类别/性质：**专业课程/必修课

**2.学分/学时：** 3.5/72

**理论学时：**40  **实践学时：**32

**3.适用专业：**软件工程

**4.先修课程：**计算机导论

**5.教材及参考书目：**

推荐教材：

* C语言与程序设计大学教程，李文斌，陈嶷瑛，王顶，清华大学出版社，2010/2,ISBN：978-7-302-21497-7,21世纪普通高校计算机公共课程规划教材

参考书目：

* C程序设计，谭浩强，清华大学出版社，2017，ISBN: 978-7-302-48087-7，“十二五”普通高等教育本科规划教材
* C程序设计语言（第2版·新版），Brian W. Kernighan，Dennis M. Ritchie著；徐宝文，李志译，机械工业出版社，2009，ISBN：978-7-111-12806-9
* C Primer Plus（第五版），Stephen Prata著，云巅工作室译，人民邮电出版社，2005，ISBN:7 -115-13022-1
* A Book on C: Programming in C, 4th Edition，Al Kelley，Ira Pohl，机械工业出版社，2004，ISBN: 7-111-20213-9
* C语言大学实用教程，苏小红，陈惠鹏，孙志岗，电子工业出版社，2004，ISBN:978-7-121-30923-6
* C专家编程，Perter Van Der LinDen著，徐波译，人民邮电出版社，2008，ISBN:978-7-115-17180-1
* 高质量程序设计指南——C++/C语言（第二版），林锐，韩永泉，电子工业出版社，2012，ISBN: 978-7-121-18617-2
* 程序设计实践，Brian W. Kernigham著，裘宗燕译，机械工业出版社，2007，ISBN: 978-7-111-21127-3
* C语言参考手册，Samuel P. Harbison;Guy L. Steele著，徐波译，机械工业出版社，2008ISBN: 978-7-111-23591-0

1. **课程考核方式：** 闭卷考试

**7.主要实践教学环节：**

实践教学在课上进行，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。

二、课程简介

对高级语言程序设计的学习和训练被认为是计算机科学的基本功之一，《程序设计基础》是软件工程专业的一门专业必修课。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)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。

三、课程目标

通过本课程的学习，要求学生达到下列基本目标：

（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。 使学生在学习《数据结构》课程时，不存在C语言的障碍。

（6）了解C语言的可移植性。

（7）了解大型程序设计和组织的思想和方法。

四、课程目标与毕业要求的对应关系

|  |  |  |  |
| --- | --- | --- | --- |
| **课程目标** | **对应章节** | **支撑毕业要求** | **备注** |
| 课程目标1 | 章节1 | 毕业要求7、8 |  |
| 课程目标2 | 章节2、3 | 毕业要求1、3 |  |
| 课程目标3 | 章节4、5、6、7、8、9、10、11 | 毕业要求1、2 |  |
| 课程目标4 | 章节5 | 毕业要求3 |  |
| 课程目标5 | 章节5、6、8、10 | 毕业要求1、2 |  |
| 课程目标6 | 章节11 | 毕业要求3 |  |
| 课程目标7 | 章节12 | 毕业要求5、11 |  |

五、教学内容及要求

**第一章 引言**

**主要内容：**语言与程序的关系；C语言与标准演化史；标准与实现的关系；C语言程序的编辑、编译、连接、运行程序的体验。

**基本要求：**初步理解“程序”的概念；掌握标准与实现的关系；掌握编辑C程序的方法，理解源字符集；初步理解翻译的步骤，掌握“编译”、“连接”和“运行”程序的方法；学会Visual Studio 2012的安装；能模仿例题编辑、编译、连接和运行另外一个程序。

**重点：**语言与程序的关系；标准与实现的关系。

**难点：**程序的编辑、编译、链接的过程。

**第二章 构成C语言程序的单词**

**主要内容：**详解第一个程序；程序与“单词”；一种特殊的“单词”：关键词；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；源文件、头文件、编译单元的概念；注释的作用；main函数的作用及常见形式；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。

**基本要求：**了解构成C语言程序的单词类别；了解C语言关键词；初步理解头文件的作用；初步理解“连接”的概念；掌握注释的使用；掌握main函数的作用及推荐的写法；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；

**重点：**本课程知识总揽：C语言（单词及单词的含义、用法）、程序设计。

**难点：**main函数的作用；从源程序到可执行文件的步骤。

**第三章 从问题求解到程序设计**

**主要内容：**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；程序与算法的关系；对同一个题，有多种算法，算法的好坏如何评价；算法的五大特征。算法的本质是处理数据；数据与数据类型；数据类型与标识符，标识符的声明；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；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。

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

**重点：**算法与程序的关系。

**难点：**算法、程序、C语言三者之间的关系；数据类型、标识符、数据对象、内存地址之间的关系；转换说明。

**第四章 运算符与表达式**

**主要内容：**左值与右值；运算符总揽及运算符优先级；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；表达式副作用、括号运算符与表达式；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；溢出简介。

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

**重点：**运算符优先级；表达式求值分析。

**难点：**类型转换。

**第五章 控制流与面向过程的程序设计**

**主要内容：**算法及基本结构：顺序结构、选择结构、循环结构；算法流程图的画法；语句的定义与分类；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；面向过程的程序设计简介。

**基本要求：**掌握算法的三种基本结构；初步了解面向过程的程序设计；理解控制算法运行路径的方法；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；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；掌握语句的概念，熟练使用if-else、switch、for、while、do、goto等语句；掌握多重循环的执行分析方法；能设计简单的算法，并根据算法编写程序；能读懂算法流程，并根据算法编写程序。

**重点：**流程图。

**难点：**循环结构与循环语句；面向过程的程序设计。

**第六章 指针变量**

**主要内容：**单重、多重指针变量的声明；指针变量本身的存储单元与指针变量所指的存储单元；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；特殊指针：const指针、空指针、void指针；指针变量的运算。

**基本要求：**掌握指针声明符、指针变量的声明方法；理解指针变量的两个关键点：存放地址、“捆绑”一块内存空间；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；理解const指针；了解空指针及通用指针的作用；了解指针变量的运算。

**重点：**指针的两大属性：类型、存放地址。

**难点：**类型决定了指针所指向单元的大小；const指针；指针运算。

**第七章 数组**

**主要内容：**数组的声明；一维数组的存储、初始化方法及使用；二维数组的存储、初始化方法及使用；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）；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；指针的运算（针对指向数组的指针）。

**基本要求：**掌握数组声明符、数组声明的方法；掌握一维数组、二维数组在内存中的存储；掌握通过下标方式访问数组中各元素的方法；掌握通过指向数组的指针访问数组中各元素的方法；掌握字符数组与其他内置类型数组之间的细微差别；重点掌握指向数组的指针及指针数组的声明、使用。

**重点：**指向数组的指针与元素为指针的数组。

**难点：**针对指向数组的指针的运算。

**第八章 函数**

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

**基本要求：**理解“函数”与“面向过程的编程”的关系；掌握函数原型声明与原型定义的方法；深入理解参数传递，尤其是“传递地址”的情况；理解生存期与作用域的概念；了解函数指针的声明、使用；掌握typedef的各种用法；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；掌握嵌套调用、递归函数的分析方法。

**重点：**参数传递。

**难点：**递归调用；生存期与声明作用域。

**第九章 预处理**

**主要内容：**预处理器与预处理指令；文件包含：#include ""与#include <>的区别；宏（对象式、函数式、预定义）定义与宏替换；条件编译。

**基本要求：**理解预处理指令的作用及给程序员带来的好处；重点掌握#include、#define预处理指令；掌握宏定义的方法，学会分析宏替换的详细过程；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；了解预定义宏，学会使用预定义宏。

**重点：**条件编译及其作用。

**难点：**条件编译及其作用。

**第十章 自定义数据类型**

**主要内容：**枚举类型及使用；结构体类型的声明；结构体对象、结构体对象数组；结构体对象成员的访问；结构体与动态链表；共同体。

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

**重点：**结构体、结构体数据。

**难点：**typedef与结构体。

**第十一章 标准库函数**

**主要内容：**标准库与非标准库；字符与字符串处理函数；文件及相关函数；标准语言补充。

**基本要求：**掌握标准库的作用；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；掌握字符及字符串处理的标准库函数的用法；掌握内存管理方面的标准库函数的用法；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；掌握标准输入输出函数的使用方法及区别；了解标准语言补充方面的相关知识。

**重点：**文件及文件操作。

**难点：**文件及文件操作。

**第十二章 项目实践**

**主要内容：**对一个项目，阐述从需要到设计到分析到编码的全过程。

**基本要求：**利用C语言完成一个中小型项目；对课程内容有个整体的把握和更深入的理解。

**重点：**项目需求分析、设计。

**难点：**项目的编码。

六、实践教学环节

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **实验/设计 名称** | **实验/设计 内容与要求** | **学时/周** | **每组人数** | **备注** |
| 1 | 程序运行环境的安装与使用 | 学会Visual Studio 2012的安装；模仿例题编辑、编译、连接和运行一个程序。 | 2/1 | 1 | 综合 |
| 2 | C程序初体验 | 使用Visual Studio2012建立C语言项目，添 加/编辑源文件、头文件；掌握在VS2012中“编译”、“连接”、“运行”程序的方法。 | 2/2 | 1 | 综合 |
| 3 | 数据输出 | 打印菱形图案；打印松树图案；打印机器人。 | 2/3 | 1 | 综合 |
| 4 | 表达式求值 | 根据所给表达式编程求值。 | 2/5 | 1 | 综合 |
| 5 | 面向过程程序设计 | 绘制流程图；编写选择结构程序；使用冒泡排序方法编程排序数据。 | 4/7 | 1 | 综合 |
| 6 | 指针应用 | 指针变量的声明及使用；指针变量的运算；使用指针编程交换变量的值。 | 2/9 | 1 | 综合 |
| 7 | 数组应用 | 一维数组的应用；二维数组的应用；指向数组的指针的应用；数组元素的指针的数组的应用。 | 4/11 | 1 | 综合 |
| 8 | 函数应用 | 递归函数应用；动态内存分配标准库函数应用；指向函数的指针的使用；有参函数的使用。 | 2/13  2/14 | 1 | 综合 |
| 9 | 预处理 | 宏定义的应用；条件编译的应用。 | 2/15 | 1 | 综合 |
| 10 | 自定义数据类型 | 结构体应用；typedef的使用。 | 2/16 | 1 | 综合 |
| 11 | 标准库函数 | 文件操作。 | 2/17 | 1 | 综合 |
| 12 | 项目实践 | 从需要到设计到分析到编码的全过程。 | 4/18 | 1 | 综合 |

七、学时分配

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **章节内容** | **理论** | **实验** | **课程设计** | **总学时** |
| 1 | 引言 | 2 | 2 |  | 4 |
| 2 | 构成C语言程序的单词 | 2 | 2 |  | 4 |
| 3 | 从问题求解到程序设计 | 2 | 2 |  | 4 |
| 4 | 运算符与表达式 | 4 | 2 |  | 6 |
| 5 | 控制流与面向过程的程序设计 | 6 | 4 |  | 10 |
| 6 | 指针 | 6 | 2 |  | 8 |
| 7 | 数组 | 4 | 4 |  | 8 |
| 8 | 函数 | 6 | 4 |  | 10 |
| 9 | 预处理 | 2 | 2 |  | 4 |
| 10 | 自定义数据类型 | 4 | 2 |  | 6 |
| 11 | 标准库函数 | 2 | 2 |  | 4 |
| 12 | 项目实战 | 0 | 4 |  | 4 |
| **合 计** | | 40 | 32 |  | 72 |