**《程序设计基础C》课程教学大纲**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 英文课程名 | C Programming | | 总 学 时 | | 64 | 学 分 | 4 |
| 课程编码 | G226002 | | 理论教学学时 | | 48 | 适用专业 | 软件工程 |
| 课程类别（请在课程所属类别栏注明选修或必修） | 通识课程 |  | 实践  教学  学时 | 实验学时 |  | 先修课程 | 并修离散数学 |
| 大类基础课程 |  | 上机学时 | 16 | 开课学院（部） | 计算机科学与技术学院 |
| 专业课程 | 专业基础  必修课 | 其它 |  | 基层教学组织 | 计算机基础课程群教学团队 |

**一、课程简介**

C是一种结构化语言, 简洁、紧凑，层次清晰，便于按模块化方式组织程序，易于调试和维护。C的表现能力和处理能力极强，具有丰富的运算符和数据类型，便于实现各类复杂的数据结构。C既可用于系统软件的开发，也适合于应用软件的开发。学习、掌握C的基本语法、编程特点，再进一步学习面向对象的C++，就能以一种熟悉的语法来学习面向对象的编程思想。因此，C是计算机专业的重要必修基础课程。

**二、教学目标**

**2.1 课程教学目标**

1. 理解并掌握C语言的基础语法以及简单数据类型、复合数据类型（结构体）等基础知识。能够运用变量、函数、循环、分支判断等知识编制初具算法复杂性的程序，具备运用这些知识解决一定复杂程度的编程问题的能力。
2. 掌握结构化编程思想，通过函数、嵌套、递归等理论知识的学习，能够具有解决复杂算法能力的编程语言能力。
3. 掌握一种软件开发环境，掌握程序开发、调试技巧，为后续学习面向对象的C++编程打好基础。培养学生利用开发工具解决/调试工程软件开发的能力。

**2.2 课程目标与毕业要求（指标点）对应关系**

该课程支撑以下毕业要求和具体细分指标点：

【毕业要求1】能够将数学、自然科学、工程基础和专业知识用于解决复杂软件工程问题。

支撑指标点1.3 能够将工程基础和专业知识用于求解软件领域复杂工程问题

【毕业要求3】设计/开发解决方案：能够设计针对复杂工程问题的解决方案，设计满足特定需求的系统、单元（部件）或工艺流程，并能够在设计环节中体现创新意识，考虑社会、健康、安全、法律、文化以及环境等因素

支撑指标点3.1：掌握解决复杂计算机工程问题所需的程序设计语言基础

【毕业要求5】使用现代工具：能够针对复杂工程问题，开发、选择与使用恰当的技术、资源、现代工程工具和信息技术工具，包括对复杂工程问题的预测与模拟，并能够理解其局限性

支撑指标点5.1：了解实际计算机应用系统中现代工具、技术及资源的现状和使用方法

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **课程目标** | **毕业要求指标点** | | **教学环节** | | | | | | | |
| **课堂授课** | | **实验** | | **作业** | | **课堂讨论** |
| 目标1：理解并掌握C/C++语言的基础语法，以及简单数据类型、复合数据类型（结构体）等基础知识。能够运用变量、函数、循环、分支判断等知识编制初具算法复杂性的程序，具备运用这些知识解决一定复杂程度的编程问题的能力。 | 指标点3.1：掌握解决复杂计算机工程问题所需的程序设计语言基础 | √ | | √ | | √ | | √ | |
| 目标2：掌握结构化编程思想，通过函数、嵌套、递归等理论知识的学习，能够具有解决复杂算法能力的编程语言能力 | 指标点1.3 能够将工程基础和专业知识用于求解软件领域复杂工程问题  指标点3.1：掌握解决复杂计算机工程问题所需的程序设计语言基础 | √ | | √ | | √ | | √ | |
| 目标3：掌握一种软件开发环境，掌握程序开发、调试技巧，为后续学习面向对象的C++编程打好基础。培养学生利用开发工具解决/调试工程软件开发的能力。 | 指标点5.1：了解实际计算机应用系统中现代工具、技术及资源的现状和使用方法 |  | | √ | | √ | |  | |

**三、课程教学内容及学时分配**

**3.1 理论教学安排**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 章节或知识点(模块) | 教学内容 | 学时分配 | 教学要求  (应明确教学重点、难点和教学方法) | 学生任务 | |
| 作业要求 | 其他要求(自学/讨论） |
| 1 | 语言介绍 | 程序开发基本概念介绍、C/C++语言介绍  C语言程序基本结构  C语言入门例程 | 2 | **教学要求：**   1. 简单了解编程语言的概念、C编程流程； 2. 了解简单程序开发过程； 3. 学会简单上机操作。   **教学重点：**   1. 讲授软件开发基本概念及C基本程序结构和语法要点； 2. 在开发环境中演示简单C例程的调试与运行； 3. 讲解开发环境的基本使用方法；   **教学难点：**在开发环境中演示简单C例程的调试与运行，让学生理解程序运行的过程；  **教学方法：**讲授 + 演示+课堂讨论 | C程序开发过程（上机项目构建）相关练习； | 讨论：   * C语言的发展历史？ |
| 2 | 数据类型与表达式 | 基本数据类型  输入输出语句和格式控制  算术运算符与算术表达式；  赋值运算符与赋值表达式；  逻辑运算符与逻辑表达式；  关系运算符与关系表达式；  逗号运算符与条件表达式；  位运算； | 8 | **教学要求：**   1. 掌握数据类型，变量的定义与语法规则； 2. 掌握C各种运算符的语法规则及使用特点； 3. 掌握’++’,’--‘运算符及其在编程中的特殊语法； 4. 掌握输入输出语句使用方法和格式控制； 5. 掌握位运算基本规则；   **教学重点：**   1. 讲授基本数据类型，运算符的语法规则以及输入输出的语法规则； 2. 结合练习，理解运算符的运算规则，学习写C语言表达式； 3. 通过编程范例演示，引导学生利用编程环境验证习题答案。   **教学难点：**和传统数学课程有差别的运算符记号和运算规则理解。特别是赋值=，增量++，减量，模运算%等。  **教学方法：**讲授 + 习题讲解 +演示+课堂讨论 | 数据类型、变量定义相关的练习；  C运算符运算规则和表达式相关的练习； | 讨论：   * 常量类型是如何识别的？ * 运算符的优限级别？ * 自增、自减前置 和后置 如何区别？ |
| 3 | 过程控制 | 选择结构和if条件语句；  多分支选择和switch语句  循环结构设计：for循环、While循环  多重嵌套循环 | 10 | **教学要求：**   1. 理解简单程序结构； 2. 理解语句的作用； 3. 掌握次数循环控制； 4. 掌握输入循环控制； 5. 掌握循环设计（包括多重循环）；   **教学重点：**   1. 讲授程序结构，选择和循环语句； 2. 通过编程范例演示，强化选择和循环结构的理解，强化利用循环设计进行解题； 3. 引导学生利用在线平台进行编码练习；   **教学难点：**循环展开的理解，循环结构的各部分（初始化、循环控制、不变式）等，分析适用循环解决的问题和掌握循环结构的设计。  **教学方法：** 讲授+演示+在线练习 | 三种程序控制结构的专项练习；三种控制结构综合运用相关的练习； | 讨论：   * 三种程序控制结构如何用流程图来表达？ |
| 4 | 数组与指针 | 一维数组定义和使用（含排序）；  多维数组定义和使用；  指针定义和使用；  动态数组定义（使用指针）；  字符数组的使用； | 12 | **教学要求：**   1. 理解数组的含义及存储数据的方式，特别是数据的存储方式对于理解指针非常重要； 2. 掌握指针的声明、书写语法，要求对动态内存分配能够熟练掌握（使用）； 3. 对字符数组有深入理解，能够熟练使用字符数组，能够使用字符串的各种方法处理字符；   **教学重点：**   1. 数组的概念和应用，字符串及处理方法，指针的概念和具体应用； 2. 通过编程范例演示，强化数组、指针和字符串的具体应用； 3. 通过在线练习的作业布置和讲解，培养解决问题能力；   **教学难点：**数组的内存分配及各单元的地址关系，理解和掌握使用指针操作数组（地址偏移）；字符数组和字符串的概念异同；指针的定义和使用。  **教学方法：**讲授+演示+讨论+在线练习 | 数组定义和使用相关的练习；  指针定义和使用相关的练习；  数组和指针结合的综合运用的相关练习； | 讨论：   * 数组的声明、使用？理解下标运算的本质含义？理解地址偏移的概念？ |
| 5 | 函数 | 函数定义和调用；  数组作为函数参数；  指针作为函数参数；  Main()函数参数使用；  递归函数；  变量的作用域和生存空间； | 8 | **教学要求：**   1. 掌握函数的各种使用方法，特别是函数参数传递的各种方法； 2. 对变量定义域和生存空间需要特别熟悉，特别强调全局变量的缺点；    1. 掌握递归的方法；   4）\*宏定义和预编译；  **教学重点：**   1. 函数的各种使用方法，特别是要求理解函数参数传递的各种方法； 2. 讲授变量定义域和生存空间； 3. 培养函数设计的能力。   **教学难点：**正确区分函数声明、函数定义、函数使用的形式；函数参数传递的各种形式，理解值传递、引用传递；变量的作用域的理解和掌握，特别是静态变量；递归的概念和使用递归设计程序的方法；  **教学方法：**讲授+演示+在线练习+讨论 | 函数声明、定义、使用相关的练习；  变量存储类型相关的练习；  变量作用域理解相关的练习；  递归问题求解的理解和程序构建相关练习； | 讨论：   * 区分：函数如何向调用方传达信息？ |
| 6 | 结构体 | 结构体定义、调用和初始化；  结构体在函数中的使用； | 4 | **教学要求：**   1. 掌握结构体作为一种复合自定义数据类型的概念； 2. 能够定义、使用结构体，特别是在函数参数中使用结构体；   **教学重点：**   1. 讲授结构体的定义和基本应用，讲授在函数参数中使用结构体； 2. 通过编程范例演示，理解结构体的使用规范，包括结构体变量和指针作函数参数及返回值； 3. 通过例题精讲（或讨论），深入理解应用结构体解题的好处；   **教学难点：**理解和掌握定义结构体、使用结构体。  **教学方法：**讲授+演示+例题精讲/讨论 | 结构定义和使用相关的练习； | 讨论：   * 结构体与class相比有什么不同？ |
| 7 | \*文件 | 文件概念和访问方式  文件的基本读写操作过程；  文件基本使用方法。 | 4 | **教学要求：**   1. 理解文件的存储和访问方式； 2. 掌握文件的打开、关闭和基本读写操作；   **教学重点：**   1. 讲授文件相关概念和访问方式，文件的打开，关闭以及读写操作； 2. 通过编程范例演示，掌握文件的读写操作；   **教学难点：**文件的概念以及文件的操作。  **教学方法：**讲授+演示 | 文件操作相关的练习； | 讨论：   * 文件读写方式的差异？ * 文件打开失败该怎么办？ |

**3.2 实践教学安排**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 项目名称 | 学时 | 类型 | 每组人数 | 教学要求  (应明确教学重点、难点和教学方法) | 学生任务 | |
| 作业要求 | 其他要求(自学/讨论） |
| 1 | 开发环境的熟悉 | 2 | 验证 | 1 | **教学要求：**熟悉编程环境；掌握基本的简单程序开发；进行1~3个程序的搭建实践；  **教学重点：**掌握基本环境内的常用菜单功能；掌握和理解程序搭建的步骤；理解各个步骤生成的文件类型；  **教学难点：**掌握和理解程序搭建的步骤；理解各个步骤生成的文件类型；  **教学方法：**讲授+实例示范+任务驱动式教学+实践 | 上交实验成果及相关文档 |  |
| 2 | 条件分支和循环 | 2 | 验证 | 1 | **教学要求：**进一步掌握编程环境；能够熟练运用条件判断和循环解决问题；  **教学重点：**选择结构和循环结构的运用  **教学难点：**指导学生利用调试工具进行程序调试；  **教学方法：**讲授+实例示范+任务驱动式教学+实践 | 上交实验成果及相关文档 |  |
| 3. | 数组的定义和使用（含动态数组） | 2 | 设计 | 1 | **教学要点：**  1）要求学生能够正确定义、使用数组；  2）能够正确声明使用动态数组；  3）具备熟练利用调试工具进行代码调试的能力；  **教学重点：**   1. 分清数组声明、使用； 2. 动态数组的定义和使用   **教学难点：**  动态数组的定义和使用  **教学方法：**  讲授+实例示范+任务驱动式教学+实践 | 上交实验成果及相关文档 |  |
| 4 | 函数及其调用 – 普通调用 | 2 | 设计 | 1 | **教学要点：**  1）对函数及其调用能够正确使用；  2）能够理解并正确进行函数调用的参数传递；  3）具备熟练利用调试工具进行代码调试的能力；  **教学重点：**   1. 函数声明、函数定义、函数使用之间的差异； 2. 函数的参数及参数的传递；   **教学难点：**指导区分函数声明、函数定义、函数使用的形式的差别；函数参数传递方式的差异。  **教学方法：**讲授+实例示范+任务驱动式教学+实践 | 上交实验成果及相关文档 |  |
| 5 | 函数及其调用 – 递归 | 2 | 设计 | 1 | **教学要点：**  1）完成函数递归程序的开发，理解递归的含义；  2）具备熟练利用调试工具进行代码调试的能力；  **教学重点：**递归程序的原理和构造；  **教学难点：**让学生理解递归的原理  **教学方法：**讲授+实例示范+任务驱动式教学+实践 | 上交实验成果及相关文档 |  |
| 6 | 指针定义及使用 | 2 | 设计 | 1 | **教学要点：**   1. 对指针能够正确定义； 2. 能够在函数调用中使用指针；   3）具备熟练利用调试工具进行代码调试的能力；  **教学重点：**指针的概念和指针的含义、使用  **教学难点：**正确理解指针的含义和运用指针  **教学方法：**讲授+实例示范+任务驱动式教学+实践 | 上交实验成果及相关文档 |  |
| 7 | 结构体操作 | 2 | 设计 | 1 | **教学要点：**  1）能够正确定义、使用结构体，结合函数能够进行结构体参数的调用；  2）具备熟练利用调试工具进行代码调试的能力；  **教学重点：**结构体的定义和使用  **教学难点：**结构体的定义和使用  **教学方法：**讲授+实例示范+任务驱动式教学+实践 | 上交实验成果及相关文档 |  |
| 8 | 综合实验 –数组、函数调用和排序 | 2 | 设计 | 1 | **教学要点：**  1）利用函数、数组完成简单排序算法；  2）具备熟练利用调试工具进行代码调试的能力；  **教学重点：**数组应用，排序算法的原理  **教学难点：**排序算法的原理  **教学方法：**讲授+实例示范+任务驱动式教学+实践 | 上交实验成果及相关文档 |  |

注：教师可根据实际的教学情况适当调整实践教学内容，题目数量可适当增减，实验学时也可调整。

**四、考核方式及成绩评定方式**

本课程的考核强调课堂教学成果与学生上机实践的综合考核。学生成绩由考试成绩**(70%)**和平时成绩**(30%)**构成，大类课程考试成绩包括期中考试（20%）和期末考试（50%）。平时成绩主要考核学生的实验能力、作业或课堂讨论表现。

**五、教材、课程网址及参考书目**

教 材：《C++程序设计》（第3版）. 谭浩强. 清华大学出版社，2015,8. ISBN: 9787302408307 （大类教材）

《C程序设计基础》科学出版社，胡同森、田贤忠主编 （非计算机、信息学院教材）

课程在学校网络教学平台的地址(核心课程必填)： http://i.mooc.chaoxing.com

参考书：[1] C Primer Plus（第五版）中文版. Stephen Prata. 云巅工作室译. 人民邮电出版社，2005. ISBN: 7115130221

[2] 解析C程序设计（第2版）. 甘玲．清华大学出版社，2012. ISBN: 9787302276814

**执笔者：毛国红、王英姿**

**审核者：陈志扬、田贤忠**

**课程教学团队成员：陈志扬、江颉、赖梅、刘文捷、毛国红、钱能、田贤忠、王英姿**