

《计算机导论与程序设计》教学大纲

课程编号：CS006001

课程名称：计算机导论与程序设计 英文名称：Introduction to computer and
Programming

学分/学时：4/84

课程性质：必修

适用专业：电子信息类

建议开设学期：第 1 学期

先修课程：无

开课单位：各学院

一、课程的教学目标与任务

目标：

1. 在介绍计算机基础知识的同时，从电子信息类专业特点出发引导学生理解计算机的基本工作原理，从系统的角度使新生初步建立起对计算系统的本质认识，激发学生的专业兴趣，同时为学习后续课程学习作铺垫。课程中还将介绍与计算机专业相关的人文素养与工程职业道德，以及不同领域的最新技术与研究进展：包括物联网技术、云计算技术、大数据与人工智能等；

2. 通过本课程的学习，使学生学会从计算机角度思考问题，初步培养计算思维、逻辑思维和过程式程序设计基本能力，能针对基本的数值计算和非数值计算问题设计运算过程，运用 C 语言作为程序设计的工具编制结构清晰、可读性好的程序，并会设计测试用例，完成程序的测试，能撰写该程序的技术报告，具备面向过程化应用程序的独立设计及实现能力。为进一步学习复杂软件系统的开发奠定理论基础及实践基础。

任务：

1. 通过导论部分的学习，掌握图灵机的基本原理、计算机的基本组成与原理；了解计算机各个领域的发展历史、现状与发展趋势等；从系统层面了解数据结构与算法、计算机网络、操作系统、数据库等软件系统的基本常识；通过实践环节熟练掌握计算机基本操作和应用。

2. 通过程序设计部分的学习培养计算思维、逻辑思维并建立程序设计的思想，重点培养学生分析问题和使用高级语言进行程序设计以解决实际问题的能力；掌握 C 语言基础知识、C 程序结构、语法及函数库、简单算法和数据结构的基本设计方法。

二、课程具体内容及基本要求

(一) 计算机的历史、现状、发展趋势与前沿技术概述(2 学时)

主要内容:

- (1) 计算机的发展历程
- (2) 计算机技术的发展现状与趋势
- (3) 前沿技术
- (4) 计算机工程与社会、可持续发展

1. 基本要求

了解计算机发展史上图灵、冯·诺依曼等科学家的关键工作；了解各种计算机类型；了解计算机的更新换代与关键元器件技术的发展；了解计算机科学技术的发展趋势；了解不同领域最新技术及研究进展。

2. 重点、难点:

重点：计算机的发展历程及重要历史人物。

3. 作业及课外学习要求:

通过互联网阅读计算机发展相关的文章、报告；阅读新技术的介绍性文章，了解各个领域的最新技术

(二) 计算机体系结构及其编码方式(4 学时)

主要内容:

- (1) 如何使机器具有计算能力？
- (2) 图灵机与冯·诺依曼原理
- (3) 二进制表示及运算
- (4) 数值数据的编码表示及运算
- (5) 文字符号的编码

1. 基本要求

了解图灵机模型的构成和计算过程；了解计算的过程可以被分解成一系列非常简单的动作；掌握冯·诺依曼原理；掌握二进制及其运算特点，二进制与八进制、十六进制的关系，十进制与二进制的相互转换；了解 ASCII 码字符集合，了解汉字在计算机中有多种编码；能够理解原码和补码表示及运算。

2. 重点、难点:

重点：冯·诺依曼计算机的“存储程序、程序控制”概念；计算过程可以被分解为一系列非常简单的动作；计算机的语言(机器语言)是二进制，数值数据和非数值数据都需要进行编码才能在计算机中存储和处理；数据的二进制表示特点和运算规则。

难点：图灵机模型的认识，程序的概念，数据编码。

3. 作业及课外学习要求：

练习不同进制数据编码表示方法，转换方法和运算规则。

(三) 计算机组成与软件系统(4 学时)

主要内容：

- (1) 计算机硬件组成
- (2) 计算机体系结构
- (3) 机器语言、汇编语言和高级语言简介
- (4) 计算机软件，数据结构与算法、操作系统与数据库系统简介

1. 基本要求：

了解计算机硬件系统的基本组成与体系结构，了解输入/输出设备、运算器、控制器、内存储器、外存储器，各基本组成部分之间的合作关系；了解各种计算机外围设备和存储设备。

了解程序设计语言的发展，以及机器语言、汇编语言和高级语言概念；了解程序在计算机上运行前需要进行翻译，编译是语言翻译的一种基本方法。了解系统软件的功能和主要作用。

2. 重点、难点：

重点：计算机硬件组成；高级语言需要翻译成机器语言才能为计算机识别，编译与解释；软件在计算机系统中的作用以及应用软件和系统软件的基本分类情况；操作系统软件在计算机系统中的作用。

难点：计算机系统各部件的协作、编译与解释等语言翻译基本概念。

3. 作业及课外学习要求：

通过网络资源进一步加深对计算机体系结构、软硬件系统、编程语言的了解。

(四) 计算机应用 (2+4 学时)

主要内容：

- (1) Windows 视窗系列操作系统的发展历程和技术特点;
- (2) Windows 系统的设置、使用和日常维护;
- (3) Windows 环境下的网络设置和使用;
- (4) 常见 DOS 命令的使用;
- (5) 办公软件 (Word、Excel、Powerpoint 等) 的熟练使用;
- (6) 互联网、浏览器、文献检索、资料查询

1. 基本要求:

了解主流操作系统的发展历程和技术特点; 重点介绍 Windows 操作系统的发展历程和特点; 掌握 Windows 操作系统下的常用操作和设置; 掌握常用办公软件的使用; 掌握命令行操作; 了解通过互联网进行文献检索和资料查询的方法。

2. 重点、难点:

重点: 掌握 Windows 操作系统下的常用操作和设置; 掌握常用办公软件的使用。

难点: 命令行操作。

3. 作业及课外学习要求:

通过上机环节练习计算机常用设置、基本操作、命令行使用和常用办公软件应用; 学习利用网络进行文献检索的方法; 通过网络资源了解 Linux 系统特点, 学习 Linux 系统基本操作方法。

(五) 程序设计概述 (2 学时)

主要内容:

- (1) 程序设计语言发展历史及趋势
- (2) 程序设计基本概念
- (3) 程序执行的原理和过程
- (4) C 语言概述

1. 基本要求

- (1) 了解程序设计语言的发展历史及趋势, 了解不同类型语言的特点
- (2) 了解程序设计基本概念
- (3) 理解程序执行的原理和过程, 对程序在计算机上的执行过程建立基本概念
- (4) 掌握 C 语言程序的框架结构和创建简单 C 程序的方法

2. 重点、难点

重点: 程序设计基本概念、程序执行的原理和过程

难点：C 语言程序的结构和创建简单 C 程序的方法

3.作业及课外学习要求

作业：在开发环境中编译、运行两个简单程序；练习命令行使用方法

课外学习要求：通过网络搜索与程序设计相关的内容，了解与程序设计有关的基础知识

（六）数据对象与计算(4+4 学时)

主要内容：

- (1) C 语言中数据对象的表示方法
- (2) 算术运算的表示方法
- (3) C 语言程序与用户交互方法（输入/输出）
- (4) 简单数值计算问题的程序结构

1.基本要求

- (1) 熟悉 C 语言字符集、保留字集、标识符、算符等基本词法元素
- (2) 熟悉 C 语言数据类型，掌握变量声明方法
- (3) 掌握 C 语言运算符、表达式与计算过程
- (4) 熟悉语句和复合语句的概念和作用
- (5) 掌握键盘输入和屏幕输出等交互方法
- (6) 熟悉数值计算类 C 程序的一般结构

2.重点、难点

重点：基本数据类型、运算符与表达式、基本的输入和输出处理、语句、程序结构

难点：变量声明、数学表达式的表示，输入输出方法

3.作业及课外学习要求：

作业：编写 1-2 个简单数值运算的程序

课外学习要求：通过上机练习掌握数值计算类程序中的变量表示、表达式写法以及程序结构

（七）基本程序控制结构(6+8 学时)

主要内容：

- (1) 结构化程序设计概念、
- (2) 程序流程控制结构及 C 语言相关语法

1.基本要求

- (1) 熟悉结构化程序设计概念
- (2) 掌握关系表达式和逻辑表达式的构造方法

(3) 掌握分支结构及相关语法: if 语句和 switch 语句; 嵌套 if 语句和多选项决策

(4) 掌握循环结构及相关语法: for 语句; while 语句; do-while 语句; 嵌套循环; break 和 continue 的作用;

(5) 掌握多种控制结构嵌套使用方法

2.重点、难点

重点: 结构化程序设计概念、逻辑表达式、分支结构和循环结构的基本形式及执行流程

难点: 逻辑表达式; if 语句的匹配原则; 循环语句的结束条件和执行次数; 复合控制结构的流程。

3.作业及课外学习要求

作业: 编写 4-8 个使用流程控制结构的程序

课外学习要求: 通过上机练习掌握 C 语言基本流程控制结构

(八) 函数与模块化编程(6+4 学时)

主要内容:

- (1) 模块化程序设计概念
- (2) 函数定义及使用
- (3) 变量的作用域和生存期
- (4) 递归方法
- (5) C 语言常用库函数

1.基本要求

- (1) 了解模块化编程
- (2) 熟悉函数的基本概念
- (3) 掌握函数的定义、声明和调用
- (4) 掌握参数传递机制和变量的作用域及生存期概念
- (5) 了解递归方法
- (6) 熟悉 C 语言常用库函数

2.重点、难点

重点: 函数的定义、声明和调用, 参数传递, 变量(数据)的作用域和生存期, 标准库函数的使用; 模块化编程的意义

难点: 函数定义及调用、参数传递机制; 递归函数的运行过程和递归函数的编写。

3.作业及课外学习要求

作业: 将完成特定功能的程序语句编写为函数(如素数判定、求最大公约数等), 利用函数改善程序结构, 实现模块化

课外学习要求: 通过上机练习掌握函数定义方法和函数调用方法, 了解一些常见问题的

递归解决方法

(九) 数组与字符串(6+8 学时)

主要内容:

- (1) 数组的作用
- (2) 数组定义及使用 (一维数组、二维数组)
- (3) 字符数组和字符串
- (4) 基于数组的简单算法 (查找、排序)

1.基本要求

- (1) 熟悉数组的含义及在内存中的表示
- (2) 掌握数组定义及使用方法
- (3) 掌握数组相关的常见算法 (查找、排序等)
- (4) 了解数组作为函数参数的本质和使用方法
- (5) 掌握字符数组与字符串的区别以及字符数组使用方法
- (6) 熟悉常用字符串处理库函数的使用方法

2.重点、难点

重点: 声明数组和引用数组元素的语法; 数组作为函数参数的本质和使用方法; 字符串的特殊性; 字符串处理库函数的使用。

难点: 数组的含义及在内存中的表示; 数组作为函数参数; 数组相关的常见查找和排序算法。

3.作业及课外学习要求

作业: 编写 2-4 个数组相关的程序

课外学习要求: 通过上机练习掌握数组定义和使用方法, 了解一些可以用数组解决的实际问题

(十) 指针与动态存储管理(6+4 学时)

主要内容:

- (1) 指针的本质及在内存中的表示;
- (2) 指针变量定义及使用
- (3) 指针与数组
- (4) 动态存储管理

1.基本要求

- (1) 掌握指针的含义以及在内存中的表示
- (2) 掌握如何定义和使用指针引用变量
- (3) 掌握指针运算的意义以及利用指针访问一维数组元素的方法

- (4) 熟悉指针作为函数参数的含义与使用方法
- (5) 了解动态内存管理的概念
- (6) 了解常用内存管理的函数及使用方法

2.重点、难点

重点：指针的概念及其含义；使用指针变量的方法；指针与数组的关系以及指针运算的含义。

难点：指针的本质以及在内存中的表示；指针与一维数组的关系

3.作业及课外学习要求

作业：通过上机练习掌握程序运行时指针变量的变化，使用指针容易出现的错误

课外学习要求：了解不同语言的内存管理机制

(十一) 结构体与复杂数据表示(2+4 学时)

主要内容：

- (1) 复杂数据如何用结构体（含结构体嵌套）表示
- (2) 结构体定义及使用方法

1.基本要求

- (1) 掌握结构体的声明及数据成员引用方法
- (2) 掌握使用结构体嵌套来定义复杂数据的方法
- (3) 掌握结构体作为函数参数的特点及使用方法

2.重点、难点

重点：结构类型的定义和应用。

难点：使用结构体（嵌套结构体）定义复杂数据的方法

3.作业及课外学习要求

作业：编写 1-2 个使用结构体的程序

课外学习要求：了解结构体在数据结构中的应用（例如链表等）

三、教学安排及方式

总学时 84 学时，其中：讲授 36 学时，实验 0 学时，上机 40 学时，实践 4 学时，线上 4 学时。

序号	课程内容	学时	教学方式
1	计算历史、现状、发展趋势与前沿技术概述	2	线上

2	计算机体系结构及其编码方式	4	讲授
3	计算机组成与软件系统	4	讲授
4	计算机应用	2	线上
5	实验一：计算机应用实践	4	上机
6	程序设计概述	2	讲授
7	实验二：开发环境、机试平台及基本程序结构	4	上机
8	数据对象与计算	4	讲授
9	实验三：基本数据类型和计算，格式化输入输出	4	上机
10	基本程序控制结构	4	讲授
11	习题课 1	2	实践
12	实验四：程序流程控制	8	上机
13	期中考试	4	上机
14	函数与模块化编程	6	讲授
15	实验五：函数及其使用	4	上机
16	数组与字符串	4	讲授
17	习题课 2	2	实践
18	实验六：数组及其使用	4	上机
19	指针与动态存储管理	6	讲授
20	结构体与复杂数据表示	2	讲授
21	实验七：结构体及其应用	4	上机
22	实验八：综合练习	4	上机

注：教学方式包括面授和线上，其中面授包括： 讲授、实验、上机、实践、研讨五种。

四、考核及成绩评定方式

最终成绩由计算机导论成绩、期中考试成绩、平时作业成绩以及期末考试成绩组合而成。

各部分所占比例如下：

导论部分成绩 (C1)： 20%。主要考核计算机运用能力、获取整理信息的能力以及理论联系实际的能力，学生根据任课教师提出的题目，通过对计算机的操作和应用软件的使用完成对某项计算机技术的调研，形成 PPT 形式的调研报告，报告应包含图像、文字等内容，数据准确，文字流畅、排版合理，教师根据学生完成情况给出成绩。

期中考试成绩 (C2)： 30%。主要考核分析和解决问题的能力，以及程序设计和调试能力。学生根据机试系统给出的题目编写程序，通过调试得到正确结果并通过机试系统提交。最终成绩通过上机测试确定。

平时作业成绩 (C3)： 10%。主要考核对每堂课知识点的复习、理解和掌握程度。

期末考试成绩 (C4)： 40%。主要考查程序设计思想、逻辑思维、程序设计方法、程序调试能力。考试形式为机试。

过程成绩提交时间和总评成绩计算说明表

序号	成绩提交时间	名称或说明
C1	第 6 次授课后、第 7 次授课前	导论部分成绩
C2	第 12 次授课后、第 15 次授课前	期中考试成绩
C3	第 22 次授课后，期末考试前	平时成绩
C4	期末考试后	期末考试成绩
总评成绩 = $C1 \times 0.2 + C2 \times 0.3 + C3 \times 0.1 + C4 \times 0.4$		

注：上表用于说明授课过程中分项成绩提交时间，教师应在规定的时间内提交对应成绩，提前或逾期无法提交，一旦提交无法修改。

五、教材及参考书目

教材：

1. 《C 程序设计第五版》，谭浩强，清华大学出版社

参考书目：

1. 《C 语言程序设计——程序思维与代码调试》，电子工业出版社，周幸妮
2. 《Computer Systems A Programmer's Perspective (3rd)》，Randal E. Bryant, David R. O'Hallaron, Pearson Education Limited
3. 《计算机导论与程序设计》，王俊平、孙德春等，西安电子科技大学出版社

4. 《Computing Essentials》, Timothy J.O' Leary,Linda I.O' Leary (计算机科学引论影印版), 高等教育出版社。

5. 《从问题到程序—程序设计与 C 语言引论》, 机械工业出版社 , 作者: 裘宗燕

六、说明

(一) 与相关课程的分工衔接

电子信息类其他专业课的先修课程, 包括数据结构、算法导论、操作系统等。

(二) 其他说明

无

(执笔人: 万波 审核人: 张淑平)

2019 年 2 月 28 日