

期末复习课

期末考试安排：2021-12-31 8:00-10:00

► 方式：闭卷，2小时，提供草稿纸

► 试卷题型和分值

► 简答题（15分）

基本概念

► 程序分析（50分）

分析程序错误，说明原因，不需要修改程序。

判断程序结果（从main函数开始分析，流程图帮助分析）

► 编程题（35分）

难度不会超过平时的测验。

本学期主要内容

- ▶ 过程式程序设计的本质，良好的程序设计风格
- ▶ 基本数据类型、操作符（算术、关系、逻辑、?:）
- ▶ 流程控制（复合语句、选择语句、循环语句、break、continue）
- ▶ 基于过程抽象的程序设计（功能分解）、函数的调用+参数传递，递归函数，标识符的生存期和作用域
- ▶ 构造数据类型：数组、结构类型、联合（了解概念即可）
指针、引用
- ▶ 面向对象部分了解了解面向对象程序设计和过程式程序设计的不同。

课程内容回顾——过程式程序设计的本质

过程式程序设计=算法+数据结构

程序设计的步骤：需求分析、系统设计、编程实现、测试与调试以及运行和维护。

需求分析+系统设计：算法和数据结构的设计，具体问题具体分析。

课程内容回顾——基本数据类型、表达式

整型、实数、字符、逻辑、void（数据类型的混合运算）

常量（字符常量、字符串常量）、符号常量、变量

操作符：算术、关系与逻辑、？：

- 实数之间的比较，因为有可能不能精确表示，所以不能直接用“==”，可以用近似比较： $p < 1e-10 \ \&\& \ p > -1e-10$ ，替代 $p == 0$
- /：结果和操作数类型有关
- / 和 % 可以用于获取整数位数、数字等

最简单的算法：一个表达式可以完成功能。

课程内容回顾——程序流程

流程图——可以有效地帮助理解程序的流程。

编写程序前，可以尝试先画流程图

- 1、顺序流程
- 2、分支流程 (if、 if...else...、 switch/case/default)
- 3、循环流程 (while、 do-while、 for)
- 4、流程的嵌套、复合语句的使用{...}、
流程的打断 (break、 continue)
- 5、结构化程序设计

课程内容回顾——函数

在解决复杂问题时，程序设计过程是：自顶向下，逐步精化
这一过程涉及到功能分解，将程序分解成若干子功能。

子功能对应的概念——子程序

子程序的实现——函数

函数头、功能抽象——函数的调用者需要了解的信息

函数名、形参（传递什么数据）、返回类型

函数体：封装了具体的实现

函数的嵌套调用——递归（递归条件，结束条件）

递归函数：能描述出算法，就能编写出代码。

课程内容回顾——标识符的生存期和作用域

生存期：面向对象里，对象的创建可以直观地体现。

静态：**全局变量，static变量**具有静态生存期。

自动：**局部变量和函数的形式参数**一般具有自动生存期。

动态：**动态变量**具有动态生存期。

作用域：全局、文件、局部、函数、函数原型、类、命名空间

全局：具有全局作用域的标识符能在程序的任何地方访问。使用全局标识符时，若该标识符的定义点在其它源文件中或在本源文件中使用点之后，则在使用前需要**声明**它们。

局部：函数定义或复合语句中，从标识符的定义点开始到函数定义或复合语句结束之间的程序段。在局部变量的作用域范围内，**同名全局**标识符不可直接访问，可以通过::访问

课程内容回顾——函数的参数传递方式

- ▶ 值传递：形参和实参分别拥有存储空间，各有生存期；形参的改变不会影响实参；

形参定义形式（变量定义）； 实参（变量名） `void f(int x); f(m);`

- ▶ 指针传递：避免传递大量数据，形参指向了实参的空间，通过形参可以修改实参空间的数值。

形参定义形式（指针变量定义、不带大小的数组定义+数组元素个数）；

实参（指针变量名，数组名+数组大小）

如果数组元素个数可以有其他方式判断，可以不用传递数组元素个数

`void f(int * x); f(&m); void f(int x[],int n); f(a,10);`

`void f(char * s); f(str);`

- ▶ 引用传递：形参就是实参的别名。修改形参的值，就是修改实参

形参定义（引用定义）， 实参（变量名） `void f(int & x); f(m);`

为了避免指针、引用参数传递的副作用，形参可以用`const`修饰。

课程内容回顾——数组

▶ 一维数组是常见的形式

- ▶ 整型数组、实数数组需要用**一重循环**完成输入、输出；
- ▶ 字符数组可以整体输入、整体输出；（因为有结束符\0）

▶ 数组元素的访问（**不要越界**）

- ▶ 下标访问最直观
- ▶ 指针访问：动态数组
- ▶ 注意不要越界：下标是从0开始
- ▶ 字符数组有一个'\0'，strlen(str)返回的是有效字符个数
- ▶ 在字符拷贝中，要特别注意，字符串的结尾要有'\0'

▶ 二维数组用下标访问，可读性好

课程内容回顾——指针、引用

▶ 一级指针是常见的形式

- ▶ 输出整型指针和实数指针指向的内容：* 指针

- ▶ 直接输出字符指针：将指针指向的字符串输出；如果是用*操作符，是输出一个字符。

- ▶ 指针和数组：通过指针访问数组、通过指针访问动态变量（new 出来的动态变量，需要delete掉）

- ▶ 指针和结构：单链表

- ▶ 指针和函数：参数传递、返回类型

- ▶ 引用就是变量的别名，在参数传递中有副作用，用const可以防止指针和引用的副作用。

有疑问，尽管在QQ上联系我
祝各位同学期末考出好成绩！