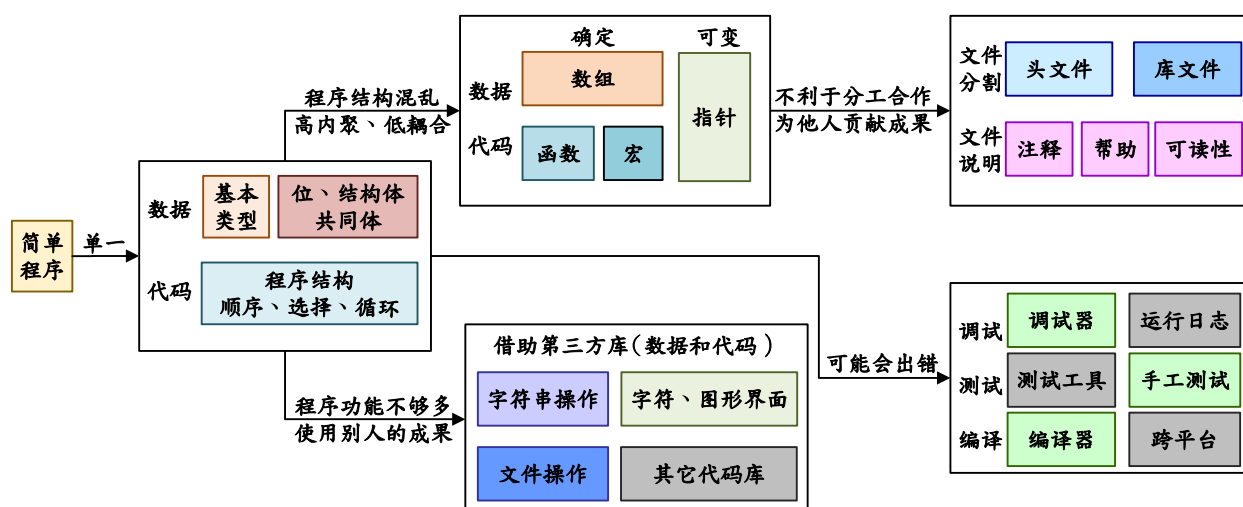


《C 程序设计》课程学期总结

本学期本班的《C 程序设计》课程兼顾深度和广度，达到较好的学习效果。在课程结束之际，我们回顾一下这个学期《C 程序设计》课程所学的内容。课堂教学难免仓促，现特整理如下，供将来计划从事软件工程相关工作的同学参考，在期末考试之后继续提高自身专业能力。

1 课程结构

我们按照以下课程结构，本质上以《C Primer Plus》为课本，讲授了 C 语言语法方面的内容。



在简单程序的基础上，我们按数据和代码两条主线一步步学习。在数据方面，我们先后了解了整数、浮点数、位、结构体、共同体和枚举等数据类型，以及有关操作；在代码方面，我们在顺序结构的基础上，学习了选择和循环的程序结构。

只用选择循环和分支书写的代码难免结构混乱，不易于围护，因而有必要进行整理，使之在数据和代码方面达到高内聚、低耦合。在数据方面，我们可以将“物理意义相同、数据类型也相同、仅有次序区别”的数据以数组的形式组织起来，并通过下标递增的方式逐一处理；在代码方面，我们可以将具有共性且不与无关内容混合的代码段封装成函数，或者将简单的代码段定义为宏，将代码模块化。对于上述数据段的数据或者代码段的内容

暂未确定的，可以定义指向数组的指针或指向函数的指针。又因为函数没有输出参量，我们学习了指针，通过指针可以直接操作内存的指定空间。

程序设计人员书写代码，难免分工合作，更不可能不与外界交流。一方面，别人设计的成熟代码，应当在版权框架内合理利用；另一方面，应吸引其他人选用自己的程序。在使用他人代码方面，我们学习了第三方库，包括：字符串操作函数、文件操作函数、字符（或图形）界面库以及其它库等。在使自己的代码易用方面，我们学习了将代码拆分成多个文件，包括头文件、源文件，并编译为库文件。并且我们要求代码编写时需要使用通俗易懂的英文名称、注释、帮助文档，符合通用的代码格式规范。

程序设计人员书写代码，不可能不出错，往往从错误到正确需要调试成千上万次程序。我们学习了断点调试（包括单步调试、断点调试）和日志调试等调试手段，学习了二分法的调试方法，测试工具和手工测试相结合的测试方法，了解制作跨平台、跨编译器的编程方法。

2 实验课

我们学习的编译方法涉及两个系统，分别是：Ubuntu 和 Windows。本课程主要在 Windows 系统上使用模拟 Linux 的 Cygwin 或 MinGW 环境。希望各位在 C++ 学习时，多接触 Windows 系统。有余力的同学可以学习下 Mac OS 系统。

我们学习的编译器涉及两大类共四种，分别是 GCC 家族：Ubuntu GCC、Cygwin 和 MinGW；以及微软家族的 Microsoft (R) C/C++ Optimizing Compiler。

我们学习的编译环境（IDE）主要是 DevC++。实际上，IDE 涉及两个操作系统、两大类，包括增强记事本类的 Notepad++ 和 GEdit；以及图形界面的 Visual Studio、Eclipse 和 Qt 等。其中微软家族的 VS 只运行于 Windows 上；而 Eclipse 和 Qt 提供了不同版本分别可以运行于 Windows 和 Ubuntu 上。VS 调用的是 Microsoft C 编译器，Eclipse 调用的是 GCC，而 Qt 则提供 MSC 和 GCC 两种版本。有余力的同学可以学习下 Mac OS 系统上的编译器。

我们学习的断点调试器涉及两种。我们在 DevC++ 中使用图形界面包装后的 GDB，它属于 GNU 家族，建议了解有控制台界面 GDB。我们还介绍了微软家族的 VS Debugger（VS 自带）。

我们学习了一种调试方法：二分法，即：每次在两条语句的中间位置查找错误，用于确定错误在该段程序的哪个部分，该法可以将错误范围缩减一半。

我们学习了一种日志调试方法，即：结合使用 `#ifdef` 和 `printf`，在关键位置输出中间结果，可以检查问题在哪一次循环。有余力的同学可以学习下 `log4cxx` 日志调试库。

我们学习了一种测试方法：等价类法、边界值法，等价类法先界定测试数据取值范围，划分等价类，对每个类选取一个代表性数据进行测试。边界值法将测试数据取值范围中的边界，相互组合成测试数据。将上述数据对程序做测试。我们还学习了利用断言测试程序的方法。

我们完成了本科专业课第一个大作业，做了一个 2048 游戏，包括：8 个基本功能和 2 个以上的附加功能，将所学循环、分支、数组、函数、文件和结构体等融合成一个整体。我们的作业提供了示例代码供大家参考。通过示例程序，我们强调了代码阅读与测试的重要性，强调程序格式和命名，强调高内聚低耦合。有些同学做出了界面等功能，有些则完成了基本功能。鼓励大家有时间继续完成剩下的历程。

我们讲解了 2 个 C 开源项目代码：Tinyhttpd 和 HUSTOJ 的 Judge_Client。有余力的同学可以学习我所推荐的其它开源项目代码，甚至阅读 Unix 早期版本的核心代码。

我们使用 MyOJ2 系统。在 3 个月期间，全班共提交 1 万余次，准确率约 58%，解决 5941 人次，每人每日平均 1.5 题。其中编译错误约 1%。答对题代码量约 360 万字符，人均约 8 万字符。

3 后续课程初步

我们涉及了两项软件工程课程的概念：高内聚、低耦合，并且在不觉之间进行了一次原型模型开发。

我们涉及了四项数据结构与算法课程的概念：链表、字符串、排序、简单查找。我们涉及了算法中常用的时间复杂度和空间复杂度概念。希望各位继续提前学习数据结构和算法。

我们涉及了一项面向对象编程的概念：类的初步。我们没有直接涉及类，而是称之为自定义数据结构，即通过数据和相关操作定义新的数据结构。它们是类定义中的成员和方法的雏形。

4 学术活动和学科竞赛

我们共同经历了 C 语言积分赛的 2 次初赛和 1 次决赛，报名人数占全班的 82%，入围决赛人数占全班 29%，获奖人数占全班约 9%。经过参赛同学的坚持，我们获得 1 项一等奖、1 项二等奖和 2 项三等奖。

我们同学组织参加了 2 个“大学生创新创业训练计划”，涉及近十位同学。

5 课程核心

我们的课程从一开始到结束，始终强调：

第一，“语文第一位、数学第二位、英语第三位、专业课程第四位，其它课程也很重要”。这句话并不是说专业课不重要，而是希望各位比专业课更重视数学和英语的重要性，尤其是语文的重要性。表达能力的重要性，在期末考试已显现出来了。在你们以后的生活中，它也将是第一位的。

第二，“阅读”与“测试”的重要性。我们小时候学习家乡话，靠的不是语法书，而是模仿家长的发音，这就是“阅读”。我们还靠我们讲错话时家长指出错误，这就是“测试”。通过阅读，见贤思齐，代码会越来越成熟。通过测试，补缺补漏，代码就会越来越严谨。希望各位继续发扬精神，重视阅读和测试的基本功，夯实基础，继续前进。

第三，“需求分析、程序设计、程序实现和测试调试是一个整体”。很多同学会有误解，我们的课程讲的是 C 程序设计，程序设计很重要。可是，不重视需求分析的后果是读

错题，浪费时间解决无用的问题。不重视测试调试的后果是做出来的代码坑很多，没有办法顺利使用。希望各位继续重视这个整体，写出高质量代码。