

很荣幸能和大家分享学习 c++过程中的心得，这是一门挑战与收获并存的课，在这里可以遇到非常认真负责的陈伦德老师，学到不止于 c++的广泛的知识，而课程深度可能会比计院更深。下面是一些经过这一个学期的感悟,对于考试，平时学习，以及项目的总结与体会。

### 一、关于复习与考试

我对试卷最直观的感受是题量很大，我在期中考和期末考都没有写完所有题目。所以在完成前面的一些基本题目时（构造函数，getters, setters）要抓紧时间，不用特别纠结卷面整洁或代码的美观，期中考我就因为总是想把卷面写的好看一点导致花了非常多的时间。审题方面，因为法语的题目通常很长，我的做法是一边读题一边用不同的符号画出关键字（例如：类名，函数名，参数类型，返回类型，功能等），防止自己遗漏。

对于卷子的后几小题，会有对于知识很灵活的应用，例如迭代器之类的比较复杂的知识点。其实我在考试的时候也没有全部完成，时间实在太紧张了。这些题目的复习可以多参考陈伦德老师的复习课以及提供的复习建议（在 gitee 里有专门的一个板块）。

考前模拟题我认为不用做很多，因为简单的题目都比较千篇一律，比较难的题又基本不会在考试里做到原题，所以它的意义在于提前熟悉一下考试题干的词汇，以及了解一些文字简答题的知识点。要强化一些复杂知识点的话还是要更多的从平时的 TD 或者复习课中学习。

### 二、关于 TD 与平时学习

每个人都可以有自己的学习方法，但比较重要的就是，要及时跟课。如果落了很多进度，想在一段时间突击补回来，对于知识的掌握是很不牢固的。长时间看视频而不自己实践代码很容易走神疲倦，而且非常容易遗忘。可能辛苦了 40 分钟，但是过几天就一点也想不起来自己到底学了什么了，到时候还得重头再来。应对遗忘，除了多实践以外，记笔记也很重要。可以传统的文字记录，也可以用代码的方式体现知识点，并在后面加上自己的注释。

平时如果遇到困惑，最快的求解方式是自己写代码尝试，这个过程需要自己思考如何测试，但是可以帮助对这个知识点留下非常深的印象（拿着自己写的测试代码再去向老师提问也很形象）。其次是查找资料，book>Google>bing>csdn>lesson>baidu，面向 baidu 编程可能需要同时打开七八个链接才能找到想要的答案，而科学上网可以高效的求解问题，还能顺便学一学英语。最后，无论是自己测试还是查资料，如果还是存有疑惑，可以多多向老师提问。

TD 课是主要的知识来源，掌握 ex 的知识点很重要（看懂 ex 的题干也很重要，可以学很多词）。可以尝试着在课后自己重新写一遍 ex，也可以尝试把 ex 里遇到的知识点用到自己的项目里，都可以加深印象，Practise>book。写 LeetCode 也是一个不错的学习方法，每周找三四道简单题写写或者看答案自己敲一遍都可以帮助快速熟悉语法，还可以学一些高效的算法用到自己的项目里。

### 三、关于项目

我还是认为项目要尽量早开始动手，虽然可能会需要在学了新知识以后回头修改以前比较落后的实现方式，但是无论是最开始写的过程还是最后回头修改的过程都可以帮助自己学习知识点，而且如果五个人能合理分工，回头修改的工程量也不会很大。这比所有知识点都学完，拖到 ddl 之前顶着期末考试的压力写项目好受多了，也能感受到自己写项目的成就感与乐趣（甚至可以用到项目管理这门课的知识）。

另外一点是，不用太在意很多不重要的细节，比如控制台打印这句话的时候要不要大写首字母，要怎么设计整个界面的布局，交互时弹出的信息要顶格写还是空两格写。我觉得以上这些方面简洁清晰就 ok，不用太过在意，更多的时间可以用在功能与设计 uml 上。

最后，祝愿大家都能顺利通过 c++考核，也祝愿大家以此为踏板，接触 cs 方面更广阔的天空。Programming is addictive (and so much fun) !

如果有所疑问或者对以上内容有不同观点，也欢迎添加我的 wx 指正与讨论：meaning42