

在训练营的学习方法浅谈

关于学习方面，我也只是个人方法的摸索，谈不上经验，抛砖引玉，交流而已。

原则：注意休息，紧跟老师进度，实在不会可暂时放弃，回头弥补，但不可掉队，因小失大不划算。

C 模块

有基础的同学，自然会轻松许多，没有基础的同学也不要着急，不会很正常，不要去怀疑自己，因为这些都是需要经过训练的。

1. 熟悉语法，由于 C 只是带着大家复习，所以课程会比较快，知道函数怎么用，然后通过课后作业来检测自己是否掌握这些语法。
2. 训练计算思维，也就是编程思维。站在计算机的角度，去思考如何解决问题。
3. 课后作业，对于初学者而言是有一定难度的，做不出来不要硬想，可以问问同学或者老师，答案不重要，过程不重要，重要的是把代码写出来，想想为什么要去这样解决问题。
4. 关于书籍 0 基础的同学看看《C Primer Plus》或者《C 大学教程》(蚂蚁书)。两本书都很好，更推荐第二本。但是这里的看书指的是从同阅读，而是当工具书查，讲到哪个知识点去翻翻就好。
5. 编程基础不错的同学推荐《C 和指针》，会有收获的。
6. 训练自己的调试能力，vs 是强大的调试工具。

Linux 模块

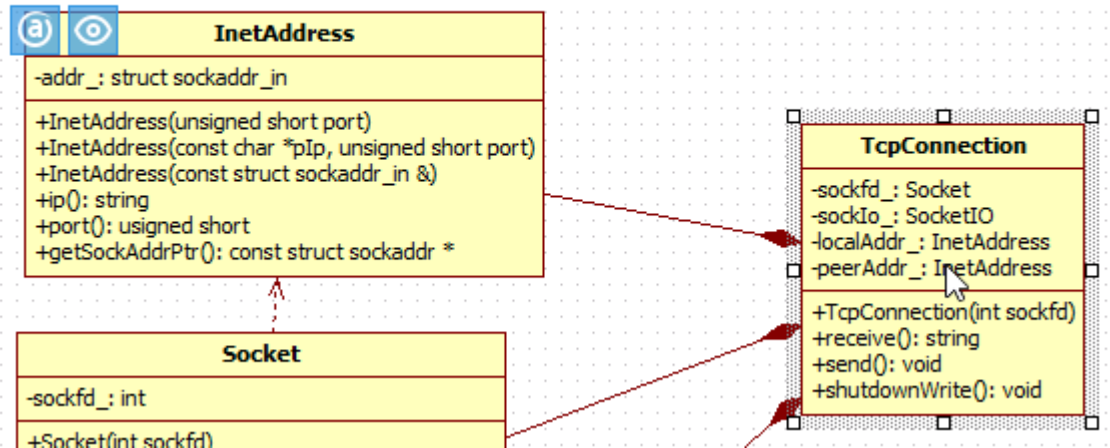
Linux 对大部分同学来说都比较陌生，不要害怕。

0. shell，命令不是重点，用两天就会发现熟悉了，不需要单独花时间，碰到需要的百度即可。

1. 课前看一下第二天的讲义，可以去往期上课的课件中找。熟悉下上课内容，重点认真听、自己困惑之处认真听。
2. 下课可以选择休息，精力不错呢就去问问上课没搞懂的地方。
3. 由于 Linux 部分，没有什么作业，所以每天要将老师的上课写的代码敲完。当然敲之前要建立在已经搞懂的基础之上，不懂就问，不要犹豫，还有就是千万不能欠账，如果欠账了，一定要补上。
4. 上课听课学会做笔记，个人建议，安装一个印象笔记，A.记录不懂之处(记下时间、对应的知识点)，方便自己对着视频快速定位。 B.记录重点(截图即可)。 下面两张图是简单的示范：

笔记本	更新时间	标题
课堂笔记	2015/11/14 8:04	2015.11.09
课堂笔记	2015/11/7 18:02	2015.11.07
课堂笔记	2015/11/6 21:58	2015.11.06
课堂笔记	2015/11/6 10:18	2015.11.05

3. readn 和 writen函数实现 避免大文件数据丢失 readline () 函数，读取一行



5. 推荐书籍《Linux 程序设计》、《Unix 环境高级编程》都是工具书，以前者为主吧，后面一本需要时间去消化。

C++模块

C++知识点多，相对于前两个模块而言，突然进入 C++可能会非常的不适应。解决之道是多敲代码，每敲一遍会有多一遍的体会。

1. 因为知识点的繁多，预习工作不可或缺。预习方法：浏览讲义、熟悉知识点、做好记录---重点听、最好是看一遍讲义代码，明白程序的目的是什么。这样的预习，好处自不待言。

2. 笔记类似 Linux 模块，学会截图，学会记录不懂得知识点、时间点。这样就不会错过重点和疑难。

3. 下课，若时间允许，就回顾一下上课知识点，对，知识点即可，不懂赶紧问。

4. 中午，若时间允许，敲一下上午代码，复习一下上午的重点，或者去问问同学自己上午没搞懂的地方。总之，别欠账。

5. 晚上时间，敲课堂代码、做作业。敲代码优先级高于作业，作业可以讨论，不要硬想，这样效率才高。

6. 实在不会，一行行写注释，先搞懂程序目的，理解函数用法，理清程序顺序，各个模块分别是如何调用的。

7. 书籍推荐《C++ Primer》，当工具书查。一本足以，切勿贪多。

8. 若上课实在听不懂，不要硬听，可以根据今日讲的内容，去书上的对应章节阅读，避免低效的学习。

关于项目

1. 项目是必须要做的，不会可以问老师，同学，当然也欢迎问我。
2. 做项目的目的： A. 帮助复习、巩固、查漏补缺 B. 锻炼自己的编程思维、调试技巧 C. 面试有的谈
3. 一共四个项目，每个项目都是小小的挑战。会有一定难度，但都是可以做出来的。

其它

1. 身体是革命的本钱，我们那一期有同学因为着凉、感染而耽误了学习，照顾好自己。
2. 大家如有不懂欢迎找我，一起交流，共同进步。
3. 要有信心，编程是要经过训练的。靡不有初，鲜克有终，共勉。