

1. C++开发训练

1.1. 基础背景

本科就读于广州大学计算机与网络工程学院的网络工程专业，除了学习计算机专业基础课程外，主要学习的编程语言是 Java 和 C++。在本科阶段，我做过的的一些项目主要使用 Java 实现，没有用 C++做过项目。

我通过学习黑马程序员的视频课程、《C++ Primer》以及侯捷的《STL 与泛型编程》来提升 C++技能。我对 C++的核心语法和特性有了一定的掌握，包括面向对象编程、模板以及 STL，此外还了解了一些内存管理和指针操作的基础。

1.2. 学习过程

我的学习过程主要分为三部分：插件学习、工具学习和 C++学习。

插件学习部分，主要学习了 CMake、Google Test 和 Doxygen。前两个插件本科阶段没有接触过，学习起来有一定难度，主要是通过阅读网上博客和官方手册，掌握了工具的语法和基本使用，并运行了一些样例。

工具学习部分，主要学习了 Git 的使用和 MobaXterm 的使用。Git 的学习主要参考了廖雪峰的 Git 教程：<https://liaoxuefeng.com/books/git/introduction/index.html>。同时，我还学习了使用 MobaXterm 连接远程服务器。

C++学习部分，主要阅读了《C++ Primer Plus》和《Effective Modern C++》这两本书，并复习了本科记录的一些笔记。

1.3. 学习成果

这是我的学习笔记：<https://docs.qq.com/s/cwJ-C3Pxcn5nIHdCRDLNYW>

在三周的 C++学习中，学习了 modern C++的新知识，这是以前没有比较陌生的内容；还有就是学习了工程化思想，在本科阶段，对变量、函数、类的命名比较随意，很少使用命名空间和移动语义这些知识，但是看了 iEDA 项目的部分代码和 iEDA 的编程规范后，发现命名有严格的规范，同时代码都比较模块化；我学习到了当要实现某个功能时，必须满足代码规划，能很好嵌入进项目，同时要有可扩展性，这也是我以后要注意和学习的地方；

1.4. 总结和建议

仅仅看书的话效率太低，还是得多实践多写代码；学习新知识点时，最好及时做笔记，否则很容易遗忘；

2. EDA 学习和实践

2.1. 基础背景

本科阶段没有接触过芯片设计和 EDA 相关内容，可以算 0 基础；

2.2. 学习过程

EDA 的学习过程是从点到面，首先大概学习了逻辑综合、芯片后端设计的各个流程(布图、布局、时钟树综合、布线等)，再把整个流程串起来，了解整个后端设计流程；根据这个后端流程，去 iEDA 项目跑 flow，了解每个流程的输入输出文件和对应的评测指标；

EDA 的学习资料主要以《超大规模集成电路物理设计》和《数字集成电路物理设计》这两本书为主，前者有涉及到布局布线等流程的一些算法，能够学得更深入一点，后者涉及芯片设计相关工艺，没有芯片基础看不太懂；

同时看了 iEDA 官网的知识和训练栏目的 EDA 知识，帮助自己认识和理解；

2.3. 学习成果

做了两本书的学习笔记；对 EDA 这个方向有了更清楚的认识，掌握了一些布图、布局和布线相关的算法，了解自己以后要在项目做的工作；

2.4. 总结和建议

在学逻辑综合入门的时候，感觉挺困难的，涉及比较多本科阶段没有接触过的概念和知识，学习计划上只安排了一天时间，很难看那么多的资料，最后只学习了逻辑综合的基础概念，没能深入到细节；

我觉得水滴计划在 EDA 这个模块的安排比较紧凑，每天的学习材料很多，讲座也很多，时间不太够用，导致学习有些仓促；希望可以精简一些学习内容；

3. 选学部分（AI/Rust）

3.1. 基础背景

本科有上过人工智能导论和机器学习的课，对深度学习有一些了解，对 AI 挺感兴趣的；

3.2. 学习过程

还在学习中，目前在看李沐的课程《动手学习深度学习》；

3.3. 学习成果

3.4. 总结和建议