# 计算机导论与程序设计课程学习报告

20049200492

马晨阳

2020.10.12

为了进一步了解计算机工作原理，更好的理解与应用计算机，掌握用计算机处理问题的方法，培养计算思维，提高分析问题和解决问题的能力，我们学习程序设计。而c语言功能丰富，表达能力强，使用灵活方便，目标程序效率高，既具有高级语言的优点，又具有低级语言的许多特点，既适于编写系统软件，又能方便地用来编写应用软件。

如何更好的设计程序关键在于如何处理好算法和语言的关系。算法与语言相辅相成不可分割。算法是灵魂，不掌握算法编程就是无米之炊。语言是工具，不掌握语言，编程就成为了空中楼阁。其次要注重培养科学思维方法，大学计算机教育要综合考虑的三个方面的因素：信息技术的发展，面向的应用需求，科学思维的培养。

至于如何在现实中程序设计，大一新生处于入门阶段，大家都说计算机入门最难，深有体会，前两节课可以说是坐听天书，接下来是朦朦胧胧，一知半解。老师讲的，关键是不知道他在讲啥，老师编的，也看不懂他在编啥。带着一张懵逼的脸，疑惑的心，打算今晚回去自学，当然看着书还是一脸懵逼，两脸问号，带着忐忑又激动的心找学姐答疑解惑，也不能解决所有问题，对于小白来说，难的不仅仅是不会，更是，不知道自己那不会，因为自己好像哪都不会。于是又怀着激动又忐忑的心，打开中国大学MOOC，找到自己考不上的北京大学，自主学习。躺在床上辗转反侧，思来想去，才发现最好的方法就是，操作！今天上机课，当实打实要自己动手操作写代码的时候，才能找到自己哪里不懂，对于新手来说细节上的失误很多，int后面少打了个（），英文的,打成了中文的，每一行后面少加了；等等等等，细节错误时是小白的一大绊脚石，其次就是格式问题，因为了解面较狭窄，什么样的类型要用什么样的格式，其实是一知半解的，以至于很多时候碰到了问题即使知道自己错在哪也不知道如何改正,可借助大佬帮助或者网课自学提升应错能力。

目前计算机课已经进行了一段时间,不过说实话对于学习了什么我并没有整理出一个体系，顺序程序的设计，数据的表现形式及其运算，对数据类型的判断，在基本运算中有算数运算比较运算和逻辑运算，算术运算中有优先级（先乘除后加减），同时也有运算符的结合性。同时 语言也很重要。目前学习停留在控制语句上，控制语句用于完成一定的控制功能，c语言只有九种控制语句，形式不同。1.if else 条件语句2 for（）循环语句 3while（）循环语句 4do while 循环语句 5 continue结束本次循环语句 6break 终止执行switch或循环语句 7switch多分支选择语句 8return从函数返回语句 9 goto转向语句，在结构化程序中基本不用goto语句（怪不得我基本没见过这东西）。



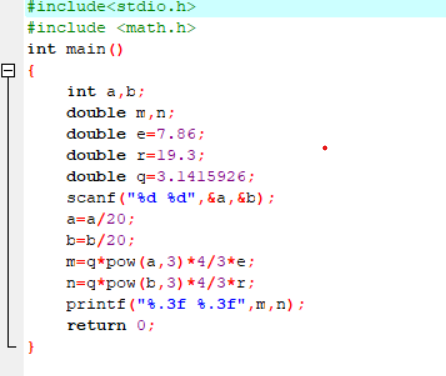
这个里面就有很多控制语句，要注意其之间的逻辑与顺序。

其次c语言的专业性较强，算法与数学有关却和而不同，在一些表达上要求较严格，像数学中的绝对值，乘方，乘号，在c语言中都有其专门的表达形。绝对值fbs，乘方pow（a，3），等等。

C语言可用于解决实际问题，可操作性强

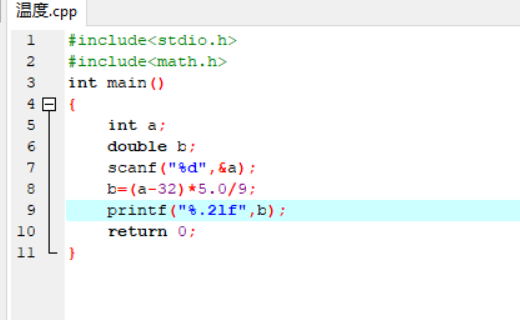
例如：问题描述：

已知铁的比重是7.86(克/立方厘米)，金的比重是19.3(克/立方厘米)。写一个程序，分别计算出给定直径的铁球与金球的质量，假定PI=3.1415926



或者是问题描述：

已知华氏温度到摄氏温度的转换公式为：摄氏温度= (华氏温度- 32)×5/9，写程序将给定的华氏温度转换为摄氏温度输出



当然这只是基础问题，还有很多很复杂的问题都可以用计算机搞定。

至于日后的学习方法，准确的来说，还没摸清楚，毕竟高中的时候花了一学期才适应了高中生活，目前老师的教学，既不写板书，也少用ppt，而是直接当堂写代码，边讲边写，很好理解却一时间难以接受，所以把老师编的拍下来下课再自己编就很ok。要想学好一门编程语言，看书远远不够，一定要把书里的代码搬到电脑里。学习编程忌讳光看不敲，看得懂不代表你会，说不定关上书或者视频，你根本无法下手

据我的了解，大部分c语言的入门者学不好c语言，都是因为一个心态问题：初学编程，费了偌大的劲，理解了这，理解了那，竟然只是在黑框下打印出了几个数字。

我觉得，初学一门编程语言一定要用合适的方式培养兴趣，黑框显然不是一个好方法。

萌新上网查一查，总结就是：就想快速入个门，跟上老师，看视频，深入学习，看书看论文看博客，最有效最实际的方法就是上机练习。

希望能在日后的日子里取得进步。不仅仅是将c语言看作一门课程，毕竟工具永远是工具，要被人玩弄于鼓掌。