第三章的学习是我认为学习C++中最重要也是最基础的一章，从这一章开始我才真正认识到学习C++的重要性。这一章的习题也是围绕着“类“来出的，通过做题，使我对类的概念、定义有了更加深刻的认识：C++中的类（Class）可以看做C语言中结构体（Struct）的升级版。结构体是一种构造类型，可以包含若干成员变量，每个成员变量的类型可以不同；可以通过结构体来定义结构体变量，每个变量拥有相同的性质。C++中的类也是一种构造类型，但是进行了一些扩展，类的成员不但可以是变量，还可以是函数；通过类定义出来的变量也有特定的称呼，叫做“对象”。结构体和类都可以看做一种由用户自己定义的复杂数据类型，在C语言中可以通过结构体名来定义变量，在C++中可以通过类名来定义变量。不同的是，通过结构体定义出来的变量还是叫变量，而通过类定义出来的变量有了新的名称，叫做对象。可以将类比喻成图纸，对象比喻成零件，图纸说明了零件的参数（成员变量）及其承担的任务（成员函数）；一张图纸可以生产出多个具有相同性质的零件，不同图纸可以生产不同类型的零件。类只是一张图纸，起到说明的作用，不占用内存空间；对象才是具体的零件，要有地方来存放，才会占用内存空间。在C++中，通过类名就可以创建对象，即将图纸生产成零件，这个过程叫做类的实例化，因此也称对象是类的一个实例。

同时，本章习题我最大的问题就是友元的理解不透彻，通过再学习，我认识到：借助友元（friend），可以使得其他类中的成员函数以及全局范围内的函数访问当前类的 private 成员。friend 的意思是朋友，或者说是好友，与好友的关系显然要比一般人亲密一些。我们会对好朋友敞开心扉，倾诉自己的秘密，而对一般人会谨言慎行，潜意识里就自我保护。在 C++ 中，这种友好关系可以用 friend 关键字指明，中文多译为“友元”，借助友元可以访问与其有好友关系的类中的私有成员。在当前类以外定义的、不属于当前类的函数也可以在类中声明，但要在前面加 friend 关键字，这样就构成了友元函数。友元函数可以是不属于任何类的非成员函数，也可以是其他类的成员函数。友元函数可以访问当前类中的所有成员，包括 public、protected、private 属性。friend 函数不仅可以是全局函数（非成员函数），还可以是另外一个类的成员函数。不仅可以将一个函数声明为一个类的“朋友”，还可以将整个类声明为另一个类的“朋友”，这就是友元类。友元类中的所有成员函数都是另外一个类的友元函数。例如将类 B 声明为类 A 的友元类，那么类 B 中的所有成员函数都是类 A 的友元函数，可以访问类 A 的所有成员，包括 public、protected、private 属性