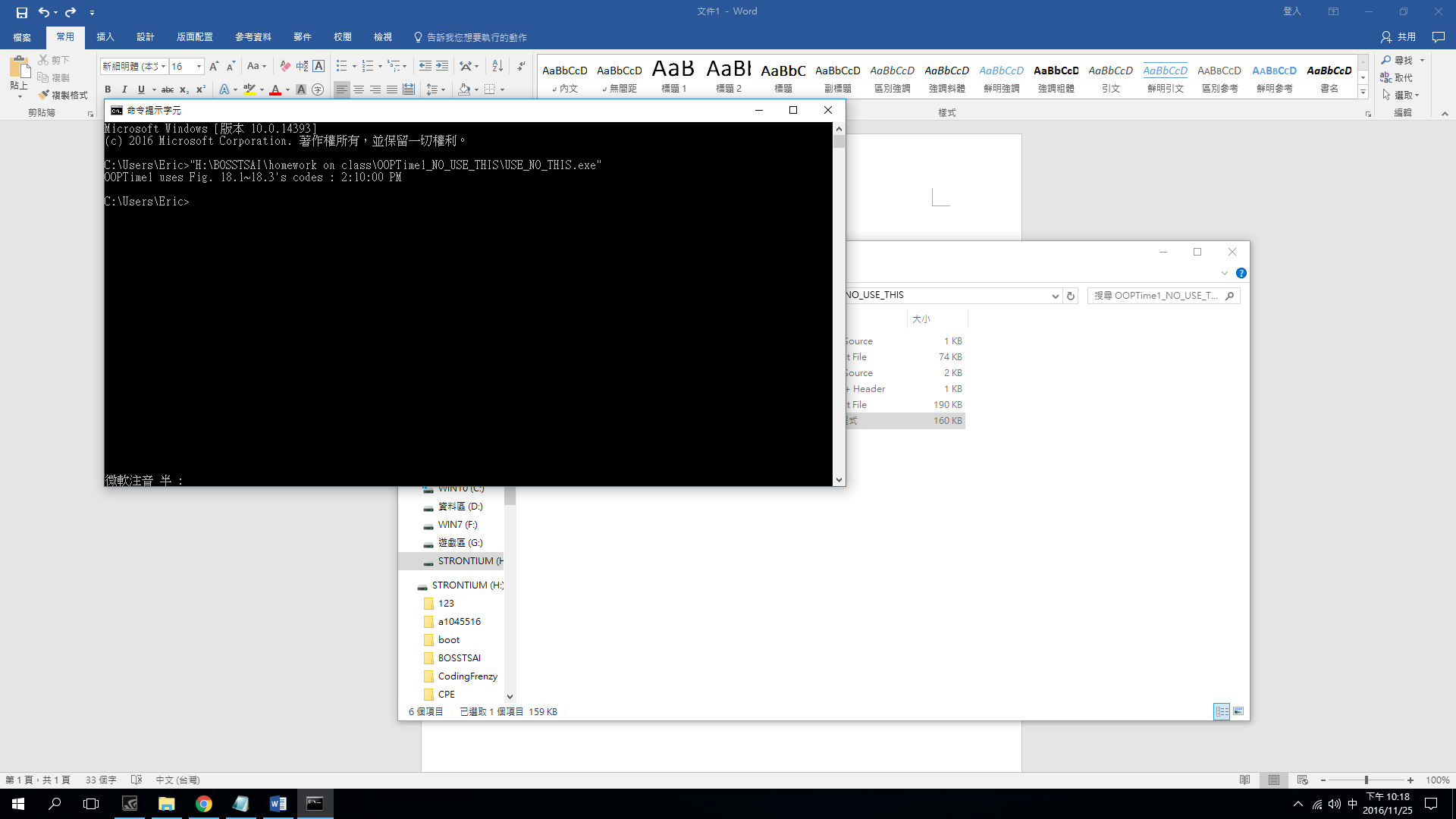
In-class practice for the this pointer

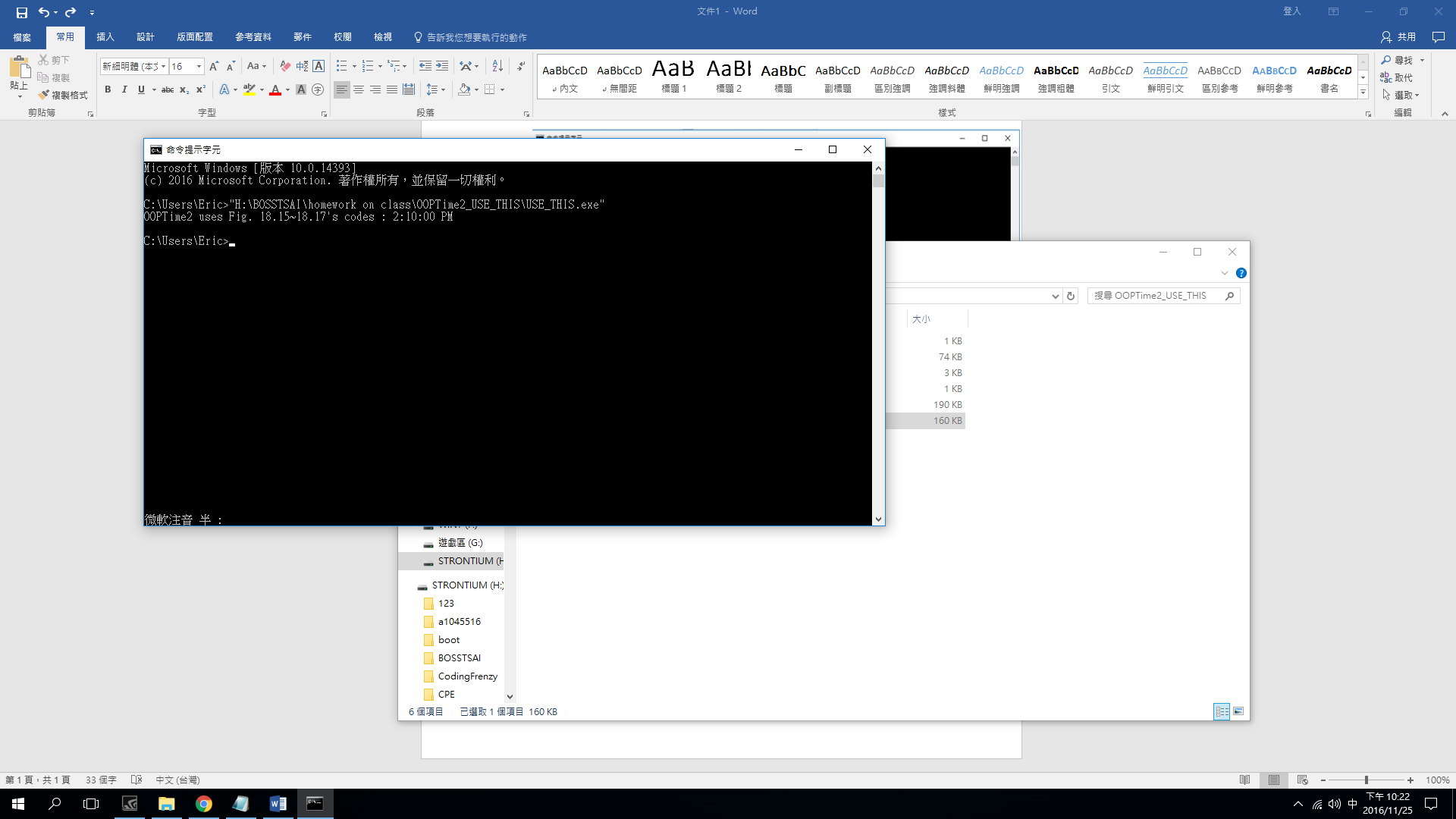
1.The results on the screen

Create two objects and print out the objects (by standard format) by the folllowing two ways:

1. OOPTime1(14,10,00): use Fig. 18.1~18.3's codes,with no using this pointer



1. OOPTime2(14,10,00): use Fig. 18.15~18.17's codes,with using this pointer



2.Discuss about the difference of the two methods of generating the objects

這兩種方法都是要去創建一個物件並且對其進行初始化的動作，而第一個方法是利用建構子來去呼叫在Time這個class當中的setTime這個function，之後再分別呼叫setHour、setMinute以及setSecond這三個function來去對Time這個class當中的hour、minute、second做初始化的動作，其形式如下:

Time OOPTime1(14,10,00);

而這一種方法是沒用到this pointer當中的Cascaded Function Calls這個方法，就只是單純的呼叫函式來去做set的動作。

反觀第二個方法就是利用this pointer當中的Cascaded Function Calls這個方法來去對其物件做初始化的動作，而這個方法的使用方式就是其function的return type 是 Time &的形式，也就是其所回傳的是一個reference，並且每一個set functions其所回傳的值必須是reference to an object，也就是function裡的return寫成這樣return \*this，那寫成這樣之後我們就能夠利用Cascaded Function Calls的形式來去初始化我們的物件，那其形式如下:

Time OOPTime2;

OOPTime2.setHour(14).setMinute(10).setSecond(00);

而上述這兩段程式碼其運作原理如下:

->首先先創立一個Time的物件叫OOPTime2

->之後再寫成

OOPTime2.setHour(14).setMinute(10).setSecond(00);

->接著，executes OOPTime2.setHour(14), returns \*this

(reference to an object) and the expression becomes

OOPTime2.setMinute(10).setSecond(00);

->然後，executes OOPTime2.setMinute(10), returns \*this

(reference to an object) and the expression becomes

OOPTime2.setSecond(00);

->最後，executes OOPTime2.setSecond(00), returns \*this

(reference to an object) and the expression becomes

OOPTime2;

那經由以上的這些步驟之後，OOPTime2這個物件也就初始化完成了。

而從以上這兩種方法也不難看出其一個很明顯的差別就是方法一的set functions並沒有任何的回傳值，其return type為void，而方法二的set functions就有回傳值，其return type為Time &，並且其回傳值為reference to an object，而這兩種初始化物件的方式可以看到其方法二的初始化方式是比較麻煩的，不像方法一只需一行就能夠去完成初始化的動作，而且一旦今天有多個物件被宣告並且要去做初始化的話，利用方法二這種初始化方法是相對比較沒有效率的，而且還有可能因為其function的名稱寫錯導致編譯錯誤的事情發生，反觀方法一則是只需在括弧中分別對每一個data member進行給值的動作，再藉由建構子來去初始化這些給定的值，這樣物件就完成了初始化，相對就會比方法二還要來的有效率。