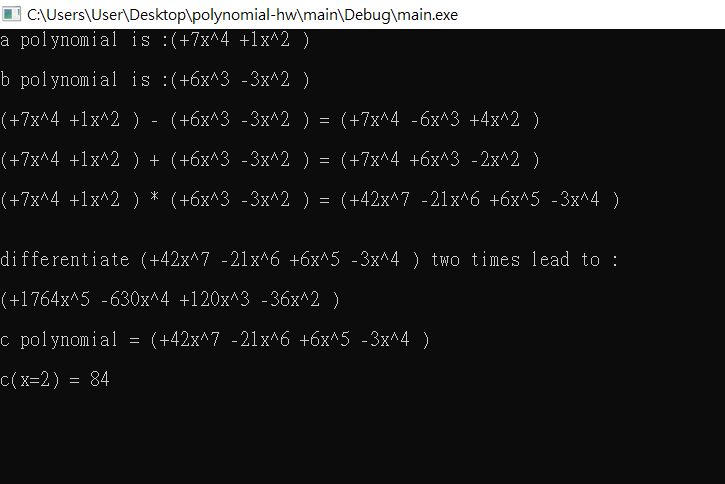
編寫一份簡短的報告，以描述您為此作業所做的事情 :

M10707617 / 梁松澤 / 2020.04.17

【Result】



2. You have to write a short report to describe :

(1) What is all about the program ?

(3) What you have done for this program homework ?

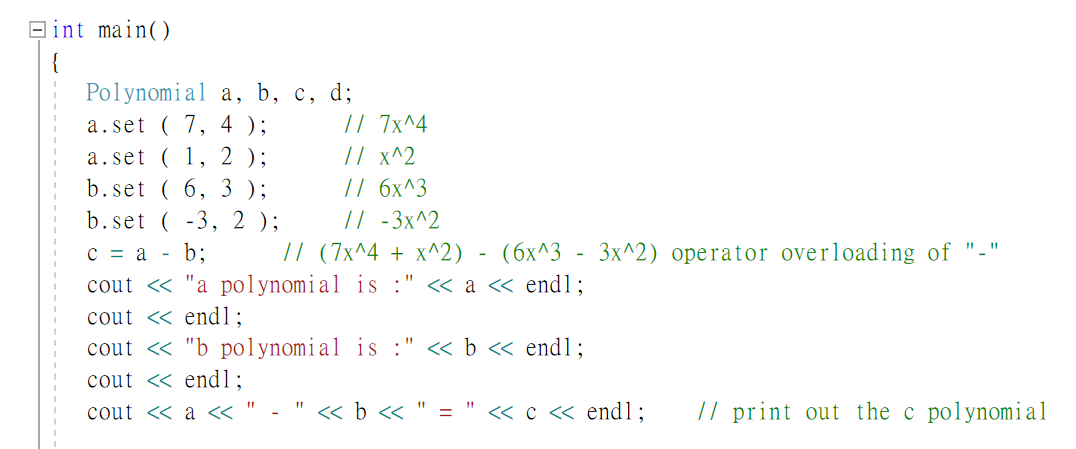
(4) How will you improve this program?

(完成 ***Polynomial 標頭檔*** 裡所有的**//do it by yourself** 的填空程式)

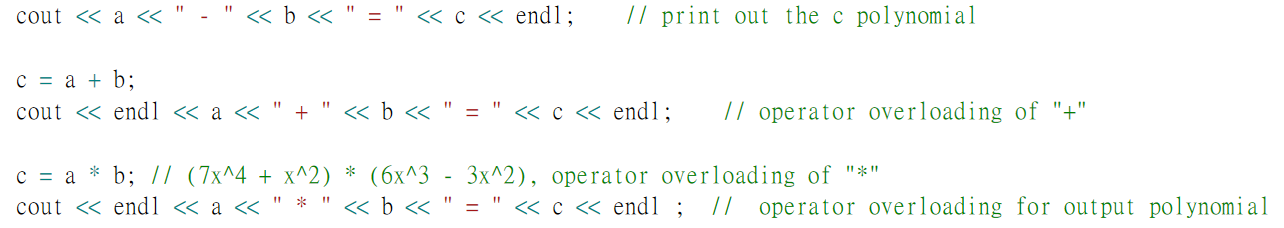
(這三題回答統一寫在一起，因為我把全部的程式碼都放在下面並都有加些小註解了)

|  |
| --- |
| In ***Main function [main ()] :*** |

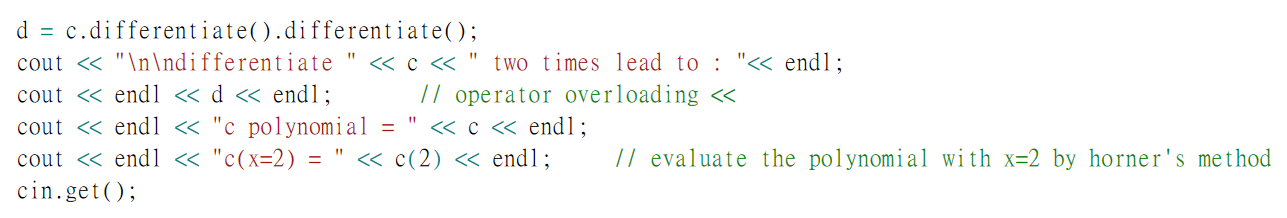
Polynomial class宣告並建立多項式 a, b, c, d 四個物件。



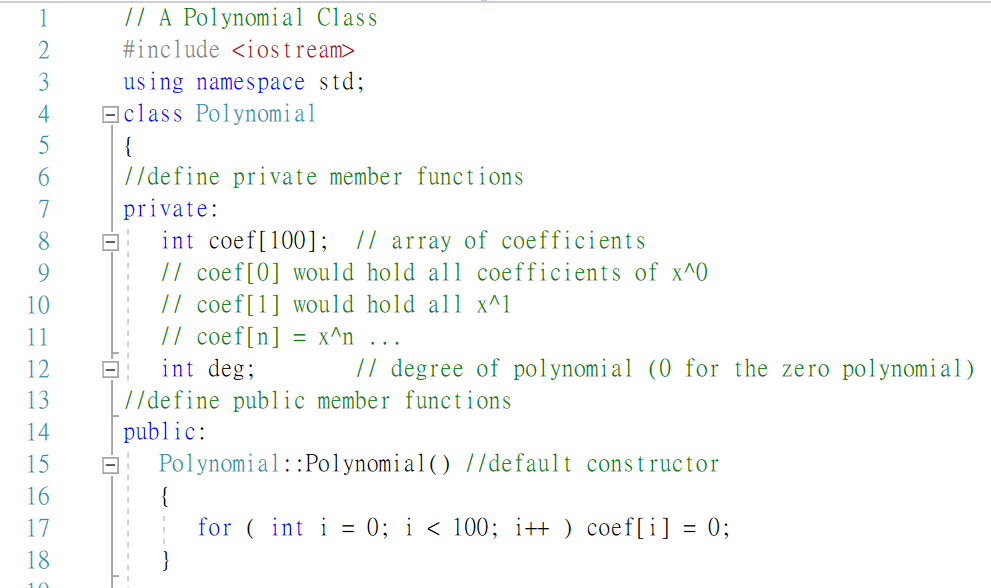
Overloading + / - / \* / ( ) 四個 operators : 使多項式可以進行四則運算，並將結果儲存在多項式 c。



對四則運算後的結果 c 進行一次和二次微分並印出。

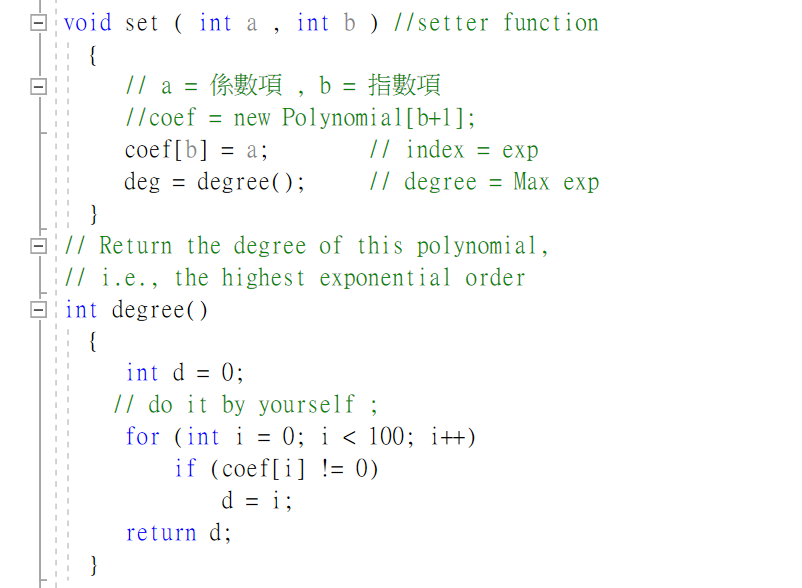


|  |
| --- |
| In ***Polynomial 標頭檔 [class Polynomial] :*** |

定義 ***class Polynomial*** 並初始化所有的 array 係數值 = 0

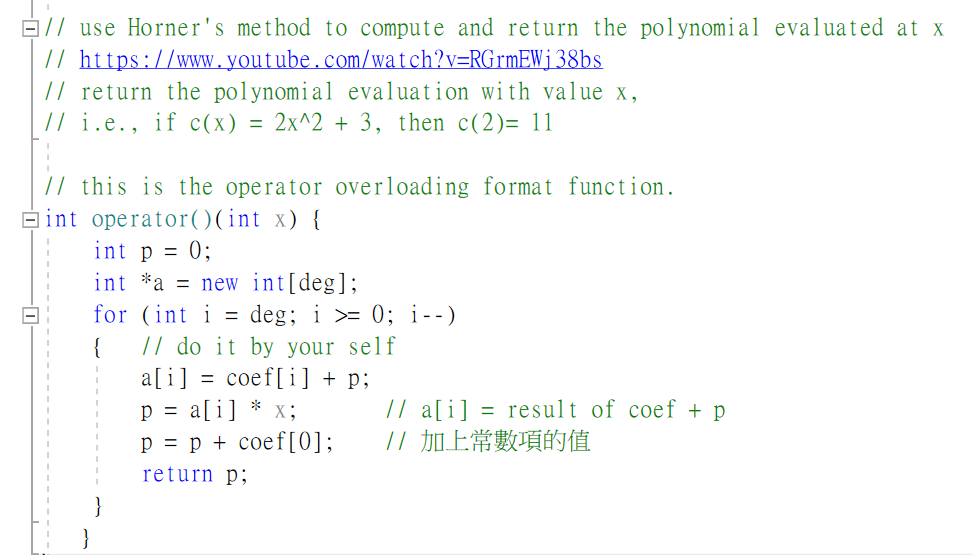
賦值給予各個 ***Polynomial*** 建立後的 a, b, c, d 四個物件以建立多項式

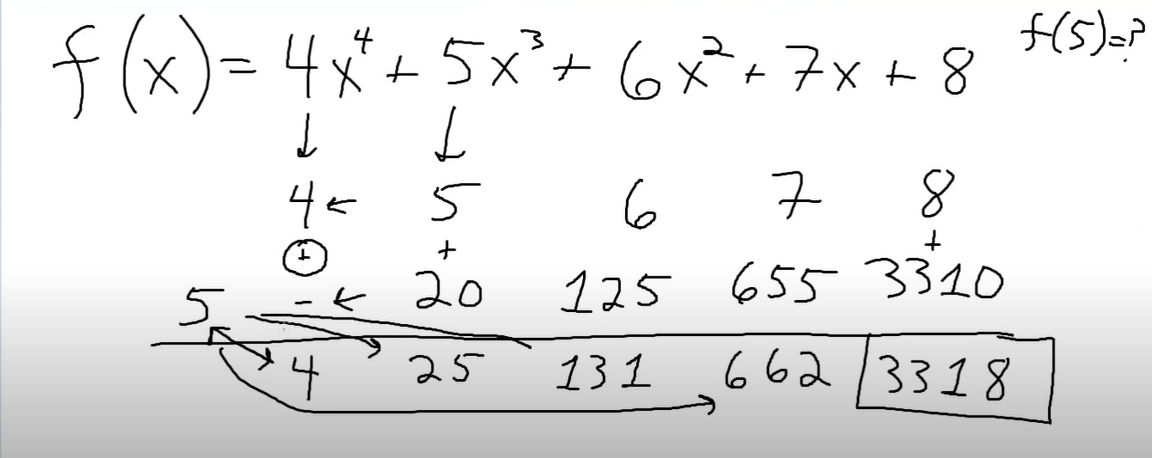
(多項式用二維 array 來儲存)



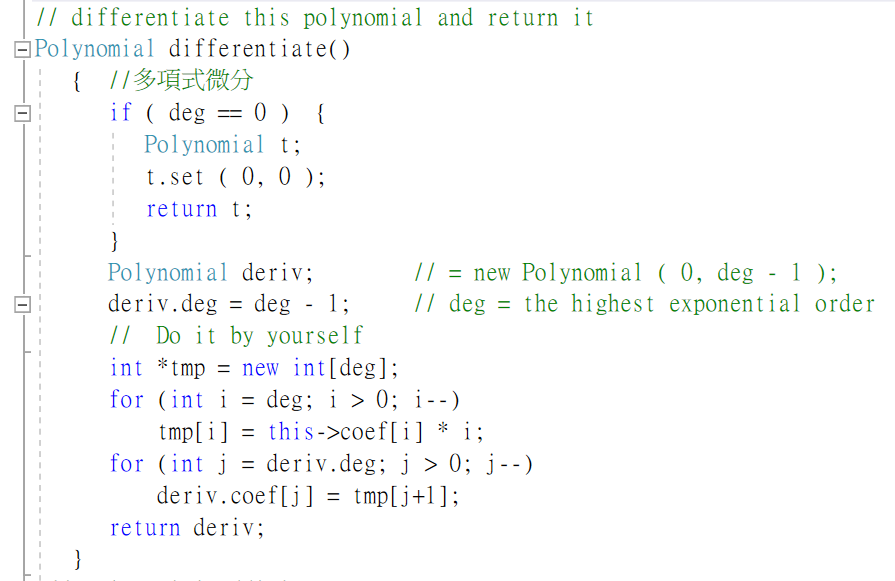
( degree 從 index = 0 開始找到最大 size 的 array，尋找途中只要係數不等於 0 就將 degree 設為該index 來代表 degree 值。)

多項式將f(x) 將 x 帶數字進去求值 : 參考 Horner’s method。

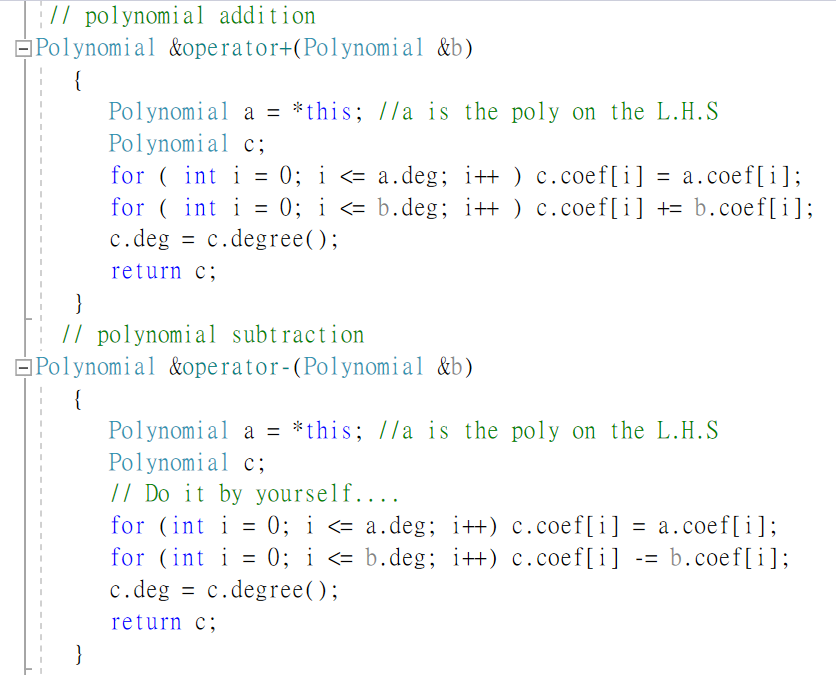




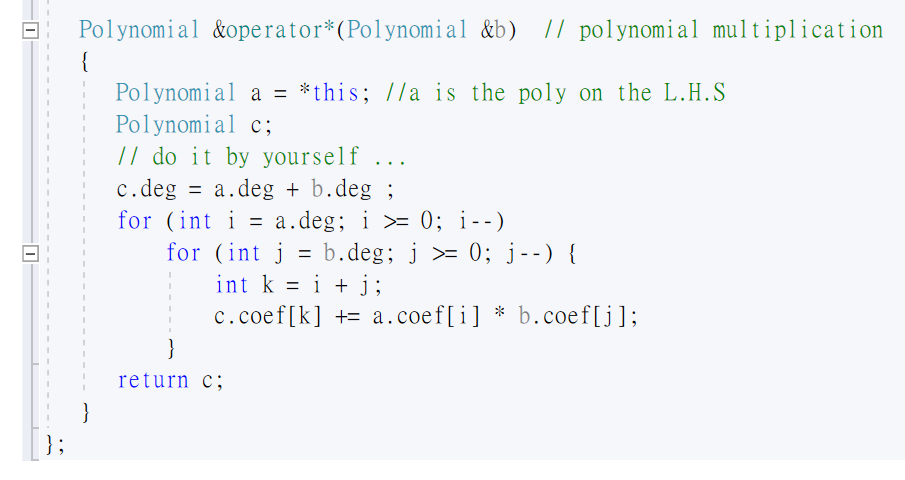
定義**多項式微分** : 記得**一定要另外宣告記憶體**用來**儲存微分後的運算值 (\*tmp)**，不然沒有配置記憶體的話，是**沒辦法直接將運算後的結果作指派或是回傳**的。



Overloading 多項式的 + 和 – 法 : 注意尤其是 – 號的第一行 **c.coef [i] = a.coef [i]中間不要用 -= 不然會導致運算結果正負號顛倒。**



Overloading 多項式的 \* 法 : 這裡注意要用三個變數去仔細思考 index的儲存邏輯。



(2) What is the time complexity reduction (%) when using Horner’s method instead of brute force one.

