Data Structure Program Assignment #3
(Due: PM: 6:00, March 28, 2025)

Polynomial class and data structure

Instructor: Jiann-Jone Chen

■ Introduction

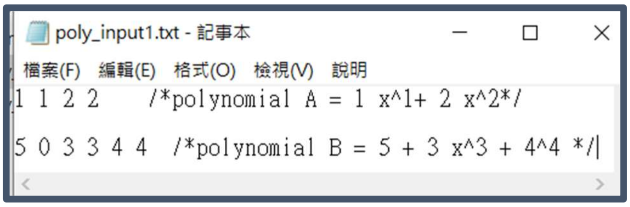
In **Lecture 2**, we introduced **three different methods** for storing polynomial data. This program is specifically designed based on **Method 1** to perform **polynomial operations**, including **addition, subtraction, and multiplication**.

Additionally, the program includes functions to:

1. **Evaluate a polynomial** for a given value (e.g., x=2x = 2x=2).
2. **Compute the derivative** of a polynomial.

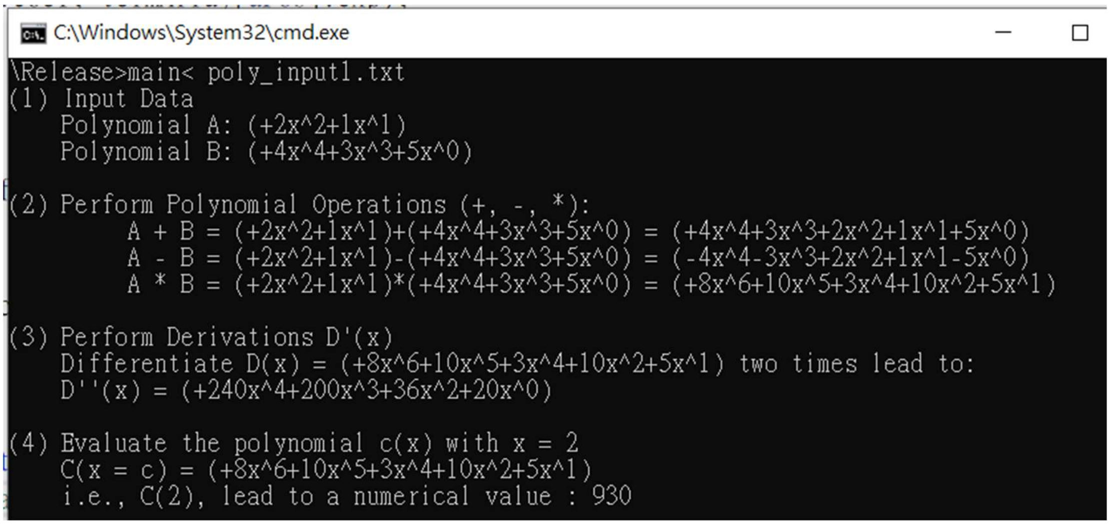
This program provides a foundational structure for polynomial operations. Your task is to **complete the implementation** so that all the instructions in the main.cpp file execute correctly using the polynomial data provided in the “poly_input.txt” file.

```
#include "Polynomial.h"
int main()
{
   Polynomial a, b, c, d;
   a.set ( 1, 1 ); //1 x^1
   a.set ( 2, 2 ); //2 x^2
   b.set ( 5, 0 ); //5 x^0
   b.set ( 3, 3 ); //3 x^3
   b.set (4, 4);    // 4 x^4

   c = a + b;   // (1x^1 + 2 x^2) + (5 x^0 + 3 x^3 + 4x^4) operator overloading of "+"
   cout << a << " + " << b << " = " << c << endl;

   c = a-b;      // (1x^1 + 2 x^2) + (5 x^0 + 3 x^3 + 4x^4) operator overloading of "-"
   cout <<a << " - " << b << " = " << c << endl;   // print out the c polynomial

   c = a * b; // , operator overloading of "*"
   cout << a << " * " << b << " = " << c << endl;   // operator overloading for output
polynomial

   d = c.differentiate( ).differentiate( ); cout << "differentiate " << c <<
     "two times lead to: "<< endl;
   cout << d << endl;    // operator overloading <<
   cout << c(2) << endl; // evaluate the polynomial with x=2 by horner's method
   cin.get();
}
```



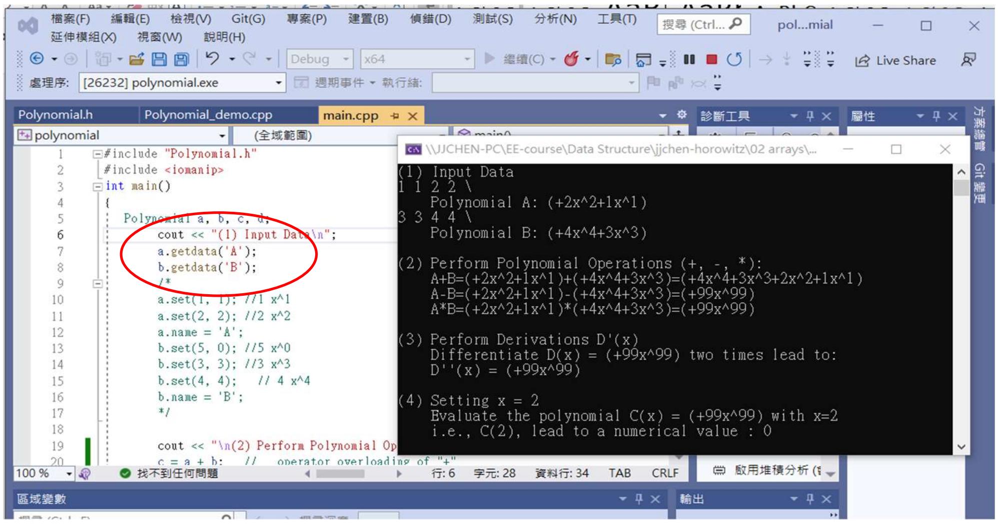
```
1 1 2 2    /*polynomial A = 1 x^1+ 2 x^2*/

5 0 3 3 4 4   /*polynomial B = 5 + 3 x^3 + 4^4 */
```

The execution results should look like the figure shown below.

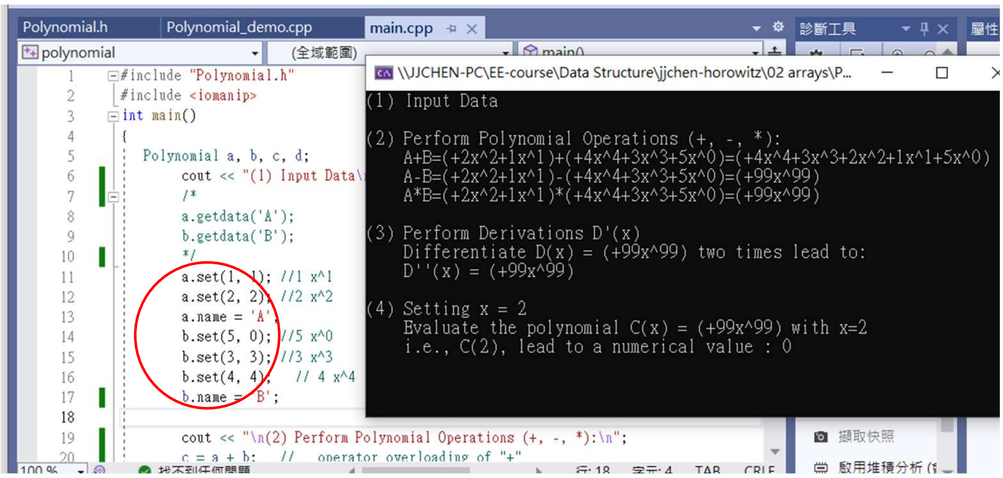


■ Steps:

1. A demo project is provided to help you quickly start the design process. Select the interactive input mode, and the execution result will appear as shown below.

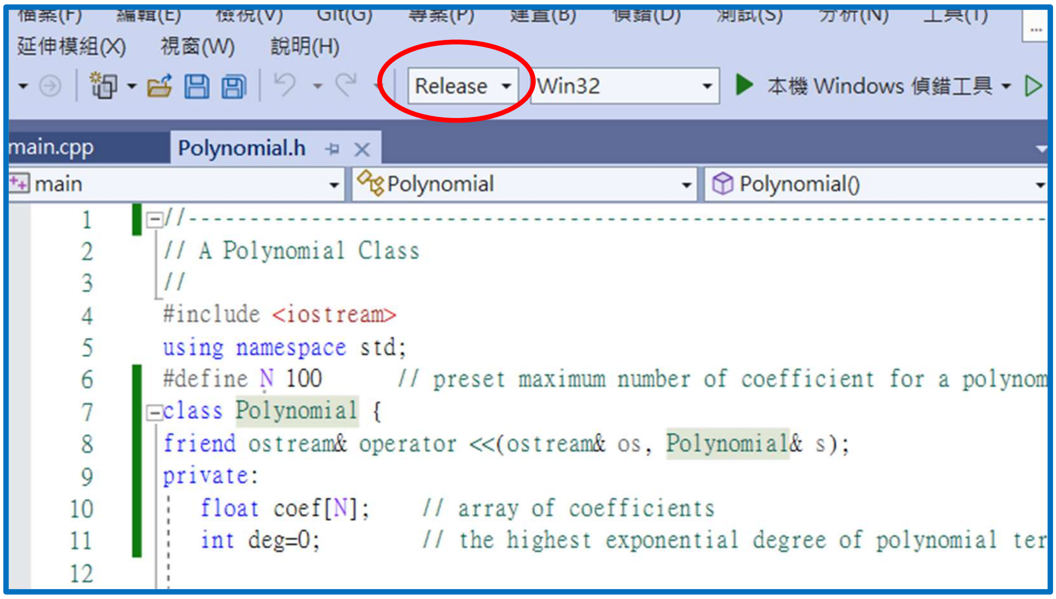


2. You can also use preset mode for easy design and debugging.

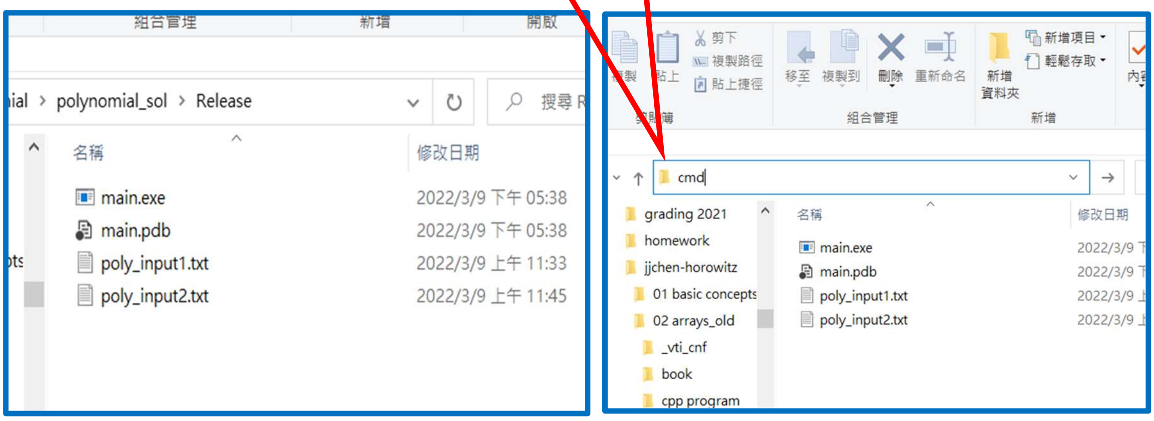


3. Design the Polynomial class to be functionally complete, enabling it to perform addition, subtraction, multiplication, differentiation, and evaluation of polynomials.

4. After completing your program, rebuild it in under Release mode



5. Go to the Release directory and trigger the cmd window.



7. Type **polynomial < poly_input1.txt** in the command window to check correctness.

```
C:\Windows\System32\cmd.exe
olynomial\x64\Release>dir
 磁碟區 G 中的磁碟是 4T_TOSHIBA_20230403
 磁碟區序號:  7AA9-81B0

 G:\WORKING_2025\Course 2024\. DS2024\02
polynomial\x64\Release 的目錄

2025/03/18  下午 04:18    <DIR>          .
2025/03/18  下午 04:18    <DIR>          ..
2025/03/18  下午 04:17            17,920 polynomial.exe
2025/03/18  下午 04:17           806,912 polynomial.pdb
2022/03/09  上午 11:33                94 poly_input1.txt
2022/03/09  上午 11:45                95 poly_input2.txt
               4 個檔案         825,021 位元組
               2 個目錄  1,799,049,809,920 位元組可用

G:\WORKING_2025\Course 2024\. DS2024\02 arrays\PHW#2 Polynomial 20220316\hw3_polynomial\p
olynomial\x64\Release>polynomial < poly_input1.tt
```

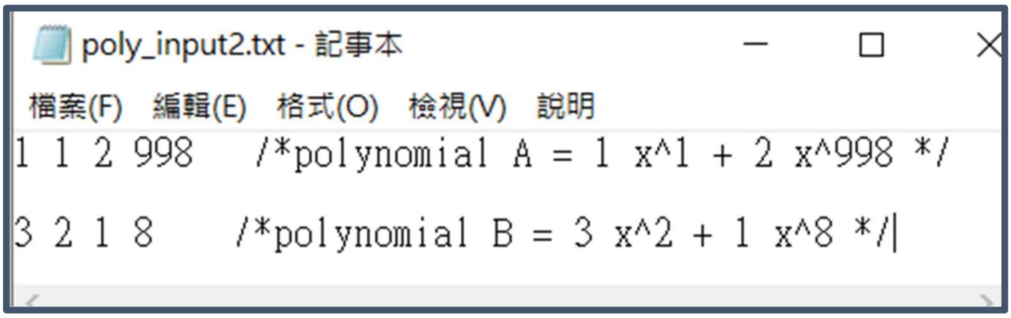
```
poly_input1.txt - 記事本
檔案(F)  編輯(E)  格式(O)  檢視(V)  說明
1 1 2 2    /*polynomial A = 1 x^1+ 2 x^2*/

5 0 3 3 4 4  /*polynomial B = 5 + 3 x^3 + 4^4 */
```

8. The result should look like the figure.

```
lease>main < poly_input1.txt
(1) Input Data
    Polynomial A: (+2x^2+1x^1)
    Polynomial B: (+4x^4+3x^3+5x^0)

(2) Perform Polynomial Operations (+, -, *):
    A+B=(+2x^2+1x^1)+(+4x^4+3x^3+5x^0)=(+4x^4+3x^3+2x^2+1x^1+5x^0)
    A-B=(+2x^2+1x^1)-(+4x^4+3x^3+5x^0)=(-4x^4-3x^3+2x^2+1x^1-5x^0)
    A*B=(+2x^2+1x^1)*(+4x^4+3x^3+5x^0)=(+8x^6+10x^5+3x^4+10x^2+5x^1)

(3) Perform Derivations D'(x)
    Differentiate D(x) = (+8x^6+10x^5+3x^4+10x^2+5x^1) two times lead to:
    D''(x) = (+240x^4+200x^3+36x^2+20x^0)

(4) Setting x = 2
    Evaluate the polynomial C(x) = (+8x^6+10x^5+3x^4+10x^2+5x^1) with x=2
    i.e., C(2), lead to a numerical value : 930
```

- **Requirements (85%)**
  1. You had to submit the complete project such that the TA can recompile your programs to verify correctness.
  2. Write a short report.doc to describe
     (1) What is all about the program?
     (2) Describe your program by writing notes for each instruction.
     (3) What is the time complexity reduction (%) when using Horner's method instead of the brute force method?
     (4) How you improve this program? List your contributions.

- **Bonus: (15%)**
  Try to enhance the program by implementing Representation 3 from the lecture notes to improve storage efficiency and operational performance. You can redesign the program to handle polynomials more efficiently, allowing it to process polynomials of any degree, including complex cases like the example below.



- You are encouraged to discuss the program design with your classmates, but do not copy code directly. If you complete the project early, please do not share your program with others. Students who submit identical program content will have to share the credit, meaning the total score will be divided among them. Make sure your submission reflects your own understanding and effort.