Project 1: Polynomial Arithmetic

Due: 11:59 pm 3/10/19

Total Points: 10

Extra Credit Points: 3 (1.5 for implementing Division and 1.5 for implementing Remainder)

Projects must be submitted on BlackBoard as a ZIPPED FOLDER with the folder name as X{8 Digit CUNY ID}

for example your student id is 12345678 than the folder name is X12345678

Within the folder will only be source code, NO .class files. The files in the folder will be:

1) X123456789.java
2) Utility.java
3) Polynomial.java
4) PolynomicalInterface.java
5) SinglyLinkedList.java
6) LinkedListInferface.java
7) Node.java
8) Term.java

Any projects submitted that **DOES NOT** have this naming convention will not be graded.

If you do not submit anything, you will receive 1 point for the project. Any projects that **do not compile or work** will receive a 0. Excuses such as “It compiles on my computer” or “It worked last time” will not be accepted. Your program must work on all machines not just yours.

If you are using an IDE such as eclipse, before submitting, remove all package statements from all files.

Late penalty Any project submitted:

1 day late will receive a max possible score of 8 and extra credit will not be rewarded

2 days late will receive a max possible score of 7 and extra credit will not be rewarded

3 days late will receive a ax possible score of 6 and extra credit will not be rewarded

Any project submitted after 3 days will not be graded.

Cheating Any one caught cheating, copying code or letting others copy, will receive a 0 and reported.

Collaborating with others is encourage on a high level, but code and implementation should never be shared.

Project Specs:

You have been hired by a math teacher to help write a program that will read in any two polynomials and will be able to do arithmetic with them. They want addition, subtraction, and multiplication for sure. If you can and have time, they would pay more for division to also be implemented.

The input will be two polynomials in string form:

“X^5+2X^2+3X^3+4X^4”

“2X^2+4X”

The output will should be:

Sum: X^5 + 4.0 X^4 + 3.0 X^3 + 4.0 X^2 + 4.0 X

Difference: X^5 + 4.0 X^4 + 3.0 X^3 -4.0 X

Product: 2.0 X^7 + 12.0 X^6 + 22.0 X^5 + 16.0 X^4 + 8.0 X^3

Quotient: 0.5 X^3 + X^2 -0.5 X + 2.0

Remainder: -8.0 X

Again the main focus is on sum, difference, and product. Quotient and Remainder are a plus.

You will be using a singly linked list to accomplish this take. Each term will be a node in the list. The length of the list will be the number of terms in the given polynomial.

You must parse each string using the Java String API (https://docs.oracle.com/javase/8/docs/api/java/lang/String.html). It is important that you learn how to parse strings and break them down into the information that you need.

You will probably spend 50% of your time parsing the strings into its individual terms.

After you parse the strings, you should insert the terms in order by power into your list. You should implement insertion sort for this part. An ordered list will be helpful when it comes to adding and subtracting.

If you do not know how to multiply/divide polynomials, please refer to a high school math text book. Or YouTube.
To check if your output is correct, you can do the math yourself, or go on wolfram alpha.

You are supplied with an outline.
DO NOT CHANGE ANY OF THE INTERFACES, UTILITY CLASS, OR THE MAIN METHOD.
You may change the two input strings for testing, but change it back to original state before you submit it.

1) Term class is used to store your term for the polynomial. It should contain both the coefficient and power.
2) The node class will hold a term object as its data
3) The LinkedList class will be a list of all the terms in the polynomial. The class implements the LinkedListInterface
4) The Polynomial Class hold the linked list that is associated with that polynomial. It implements the Polynomial interface. This is where all the arithmetic will be done. You must also implement a toString() method to print out polynomials. For now the divide and remainder method just throw exceptions. You can implement these two methods for 3 extra points. DO NOT implement these two methods until you have done the other three since division is very difficult to implement.
5) The class X12345678 will hold the main method. Like the name of the folder, please change the 12345678 to your CUNY ID.