Deadline:	Sunday November 30, at 11:59pm.
Evaluation:	6 pts of your final grade
Late Submission:	No late submission.
Teams:	The assignment can be done individually or in teams of 2 (from the same section). Submit only one assignment per team.
Purpose:	The purpose of this assignment is to help you practice arrays of objects

---

This assignment is divided in 2 parts:

Part 1

Consider a class Polynomial, which represents polynomials of a single variable up to the fourth power (x4). The sole instance variable of a Polynomial is an array of size 5, that stores doubles. Each element of the array represents the coefficients of the Polynomial at the corresponding power. For example:

Element 0 is the coefficient of the x^0 term
Element 1 is the coefficient of the x^1 term
Element 2 is the coefficient of the x^2 term
Element 3 is the coefficient of the x^3 term
Element 4 is the coefficient of the x^4 term

Thus the polynomial $3.0 – 1.0x + 2.0x^2$ would be represented by the array:

3.0	-1.0	2.0	0.0	0.0

Write the class Polynomial that includes the following methods:

1. A default constructor that initializes all coefficients to zero.
2. A constructor that takes five doubles as its formal parameters and initializes the coefficients to the value of the corresponding parameters.
3. A constructor that takes an Polynomial object as its formal parameter and creates a new polynomial that has coefficients that are identical to the Polynomial object passed in.
4. An accessor method that takes in a parameter n and returns the n^{th} coefficient.
5. A mutator method that has 2 formal parameters n and a new coefficient, and sets the n^{th} coefficient to the new value being passed in.
6. A method called **`add`** that computes the sum of two polynomials and returns a new polynomial. The signature of this method should be: **`public Polynomial add(Polynomial p)`**
7. A method called **`evaluate`** that computes the value of the polynomial with the value passed as a parameter. The signature of this method should be: **`public double evaluate(double x)`** 
8. A method called **`derivative`** that returns the derivative of the polynomial. The signature of this method should be: **`public Polynomial derivative()`**
9. A method called **`numberOfTerms`** that returns the number of non-zero terms of the polynomial.
10. A **`toString`** method such that the statement System.out.println(somePolynomialObject) would display the polynomial as an equation but only the non zero terms are displayed and the + sign for the first terms does not appear. For example:

if the array contains

3.0	-1.0	2.0	0.0	0.0

The output should be: **`3.0 – 1.0x^1 + 2.0x^2`**

if the array contains

7.0	0.0	2.0	0.0	4.0

The outputshould be: **`7.0 + 2.0x^2 + 4.0x^4`**

Part 2
Write a driver program that will :

1. Create an array of 8 Polynomials
2. Prompt the user for the coefficients for the first two polynomials (store these in locations 0 and 1 of your array of Part 1 above)
3. Randomly generates the coefficients for the next 3 polynomials (store these in locations 2, 3 and 4). The coefficicents should be between 0.0 and 100.0.
4. In location 5 of your array, stores the sum of the polynomials in location 0 and 2
5. In location 6 of your array, stores a polynomial whose coeficients are 3 times those of the polynomial in location 1
6. In location 7, store the derivative of the polynomial in location 4
7. Prompt the user for a double value x, then print the 8 polynomials along with the number of terms in each polynomial and the value of the polynomial when evaluated with x. For example if the user entered x=3.0, the output for the polynomial

3.0	-1.0	2.0	0.0	0.0

should be :
`Polynomial 3.0 - 1.0x^1 + 2.0x^2 has 3 terms and evalutes to 18.0 for x = 3.0`

Evaluation Criteria (out of 100):

Part 1 (60 pts)	
Use of significant variable names and comments, indentation and readilbility	5 pts
Declaration of the classes:	5 pts
Constructors	10 pts
Accessor method	5 pts
Mutator method	5 pts
evaluate method	5 pts
numberOfTerms method	5 pts
toString method	5 pts
derivative method	5 pts
add method	10 pts
Part 2 (40pts)	
Implementation of the methods (Driver program)	40 pts

Submission
When you are finished the classes, you must submit them online.

1) Create **one** zip file, containing both source files (the .java files).

Please name your file following this convention:

- If the work is done by 1 student, your file should be called *assignment#_studentID*, where # is the number of the assignment *studentID* is your student ID number. For example, for the first assignment, student 123456 should submit: assignment4_123456.zip

- If the work is done by 2 students: The zip file should be called *assignment#_studentID1_studentID2*, where # is the number of the assignment *studentID1* and *studentID2* are the student ID numbers of each student, for example, assignment4_123456_987654.zip
  Only one member of the team needs to upload the file.

2) Upload your zip file at the URL: https://fis.encs.concordia.ca/eas/ as ***Programming Assignment 4***