Implement a class Polynomial that models a polynomial function. You can assume that your Polynomial object has degree at most 50.

An object in the class Polynomial should have a private member variable that is an array of doubles to store the coefficients of the polynomial. (This is not a dynamic array.) If that member variable is called coeffs\_, then coeffs\_[i] should be the coefficient of the term of degree i. An object in the class Polynomial should have a private member variable that is an int to store the degree of the polynomial.

The only constructor should be a constructor with no arguments that creates the polynomial with degree 0 and all coefficients 0.

Overload the insertion operator << to output a polynomial to an output stream, including cout, in a form similar to 1.0 + 2.0\*x + 3.0\*x^2 - 4.0\*x^3. Make this operator a friend of the class.

Overload the extraction operator >> to allow for input of a polynomial from an input stream, including cin. The input should be first a non-negative integer k for the degree of the polynomial followed by k+1 doubles for the coefficients of the terms. For instance, the polynomial above would be input as 3 1.0 2.0 3.0 -4.0 Since the only constructor creates a zero polynomial, this extraction operator will be the way to actually get a non-zero polynomial. Make this operator a friend of the class. Look at the Rational class definition from Chapter 9 for examples of overloading the input and output operators.

The class should have overloaded + for addition of polynomials and overloaded – for subtraction of polynomials. Both operators must create the correct array of coefficients and the correct degree. Creating the correct degree requires some thinking. The degree of a sum can be smaller than the degrees of the polynomials begin added because terms may cancel.

Write a method eval that evaluates the polynomial at a specific value x that will be a parameter of the method. The method should return the value.

A test program that uses these methods is posted, along with file Examples.txt that is to be used to read values into polynomials. The test program will work if you put it and Polynomial.h, Polynomial.cpp, and Examples.txt in the same project. That can be done in onlinegdb .

Submit your complete class definition and the results of running the test program by the end of the day on November 23, 2022.