

This week the project consists of two parts.

For the first part, specify, design, and implement a class for polynomials.

A polynomial is an arithmetic expression of the form:

$$a_0 + a_1x + a_2x^2 + \cdots + a^kx_k$$

The highest exponent, k , is called the degree of the polynomial, and the constants are the coefficients.

The class may contain a static member constant, `MAXDEGREE`, which indicates the maximum degree of any polynomial. (This allows you to store the coefficients in an array with a fixed size.) Spend some time thinking about operations that make sense on polynomials. For example, you can write an operation that adds two polynomials. Another operation should evaluate the polynomial for a given value of x .

Please download the attached header file (`poly1.h`) and complete implementations. You may add additional functions to the header file if you wish to write more to help you complete the assignment. You may NOT alter the names or structure of any of the functions or attributes that are already within the header files. If you do add your own functions, make sure to update the header file's comments to discuss those functions' preconditions and postconditions. You will also be required to include an invariant in each of the implementation files that you write.

When doing your implementation, pay careful attention to the preconditions and postconditions given in the corresponding header file. Make sure to follow and enforce these conditions. In `poly1.h`, there are very specific instructions about how a polynomial should be printed. Make sure to follow these and ask for clarification if you are confused, as the format of what the output operator prints will be part of the grade.

For the second part, specify, design, and implement the polynomial class using a dynamic array.

Your polynomial class from the first part has the maximum degree. But in the class from the second part, there is no maximum degree.

The files that you submit should be named exactly `poly1.h`, `poly1.cpp`, `poly2.h`, and `poly2.cpp`. The namespace for the contents of these files should be carefully used. You should write test files for the classes to test all of their features thoroughly and test boundary conditions, and compare two classes before submitting your solutions.