Lab: Polynomial Arithmetic using Class Templates

Learning Outcomes

This assignment will provide you with the knowledge and practice required to develop and implement software involving:

1. Design and implementation of class templates.
2. Ability to understand and implement mathematical concepts in C++.

Polynomial Arithmetic

A polynomial in one variable is a function of the form

$$p(x) = c_0 + c_1x + c_2x^2 + \cdots + c_{n-1}x^{n-1} + c_nx^n$$

where n is a non-negative integer, $c_n \neq 0$, and other coefficients c_k may or may not be zero. The degree of $p(x)$ is n. No exponent in a polynomial is allowed to be negative. In this assignment, all coefficients are of the same C++ numerical type. Note that if the degree of the polynomial is 0, then the polynomial is just a constant.

Given two polynomials $p(x)$ and $q(x)$, the following binary operation is defined:

$$p(x) * q(x) \qquad \text{is defined to be the product of } p(x) \text{ and } q(x)$$

For example, if $p(x) = 2x + 3$ and $q(x) = -x + 1$ then

$$p(x) * q(x) = (2x + 3) * (-x + 1) = -2x^2 - x + 3$$

If you need to brush up on polynomial arithmetic, refer to any elementary algebra textbook.

`Polynomial` Class Interface

You are to declare the interface to a class template `Polynomial` that takes a template type parameter specifying type of coefficients and a non-template type parameter specifying the polynomial's degree. The class must contain the following member functions:

Name	Description
evaluate polynomial at $x = a$	Evaluate polynomial $p(x)$ using argument x and return the value. This should be implemented as an overloaded function call `operator()`.
`operator*`	Given polynomials $p(x)$ and $q(x)$, return $p(x) * q(x)$. Note that this function must only allow polynomials with similar coefficient types to be multiplied. Check your implementation with driver source file `driver-no-comp-1.cpp`.
default ctor	Creates a zero polynomial.

Name	Description	
single-argument ctor	Creates a polynomial from another polynomial. Note that a polynomial of degree m cannot be constructed with another polynomial of degree n. However, the types of coefficients for the polynomials could be different. Check your implementation with driver source file `driver-no-comp-2.cpp`.	1
copy assignment operator	Assigns a polynomial to another polynomial. Note that both polynomials must have the same degree. However, the types of coefficients for the polynomials could be different. Check your implementation with driver source file `driver-no-comp-3.cpp`.	1
`operator[]`	Define both non-modifiable and modifiable overloads of subscript operator to read and write coefficient values, respectively.	2
`operator<<`	Non-member function `operator<<` is already defined in `polynomial.tpp` and must not be altered.	X

The class template must be declared in `polynomial.h` and member functions must be defined in `polynomial.tpp`.

Submission Details

Header and source files

You will submit `polynomial.h` and `polynomial.tpp`.

Compiling, executing, and testing

Download the `Makefile`, a driver source `driver-poly.cpp` with a variety of unit tests and correct output files. Your code will be tested for memory-related problems using Valgrind. All submitted files must have file documentation blocks. The class template declaration in `polynomial.h` must further contain function-level documentation.

Submission and automatic evaluation

1. In the course web page, click on the appropriate submission page to submit the source file.
2. Please read the following rubrics to maximize your grade. Your submission will receive:
   - F grade if your submission doesn't compile with the full suite of `g++` options.
   - F grade if your submission doesn't link to create an executable.
   - F grade if Valgrind reports memory errors and/or leaks.
   - A deduction of one letter grade for each missing documentation block in submission. A teaching assistant will physically read submitted source files to ensure that these documentation blocks are authored correctly. Each missing or incomplete or copy-pasted [with irrelevant information from some previous assessment] block will result in a deduction of a letter grade. For example, if the automatic grader gave your submission an A grade and one documentation block is missing, your grade will be later reduced from A to B. Another example: if the automatic grade gave your submission a C grade and the two documentation blocks are missing, your grade will be later reduced from C to E.