ECE250: Lab Project 0

Due Date: September 13th, 2019, 11:00 pm

1. Project Description

The goal of this project is to write a C++ program that performs operations on a single variable polynomial. A single variable polynomial is a function (e.g., quadratic) involving only non-negative integer powers of x. The degree of a polynomial is the highest power of x in its expression. A polynomial of degree n is a function of the form: $f(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n$ where a's are real numbers called the coefficients of the polynomial.

We ask you to write a C++ class to represent a polynomial function and provide several services: (i) initializing the polynomial, (ii) evaluating/calculating the polynomial for a given value of x, (iii) adding two polynomials, and (iv) multiplying two polynomials.

When designing your class for a polynomial, use a linked list to store the coefficients of the polynomial. You have to write your own implementation of a linked list . It is not allowed to use the one provided by the C++ Standard Library.

2. Program Design

Write a short description of your design. You will submit this document along with your C++ solution files for marking. This document must include your design decisions. Please refer to the course website for "Programming Guidelines" and the expected content for your design document.

3. Input/Output Requirements

Write a test program named **polynomialtest.cpp** that reads commands from standard input and writes the output to standard output. This program will respond to the commands described in this section.

In the command descriptions below, we use the word "*current_polynomial", "expected_polynomial" and "second_polynomial*" to refer to a polynomial specified as a sequence like: $o\ a_0; 1\ a_1; \ldots \ldots ; n\ a_n$.

The output of each command will be either "success" or "failure".

- **Initialization and Evaluation of a Polynomial**

Command	Parameters	Description	Output
init	*m*	Defines number of coefficients. Degree of the polynomial is $(m-1)$.	success
coeff_p1	*current_polynomial*	Defines the coefficients of a polynomial function. For example, $f(x) = 1 + 10x + 20x^2$ will be defined as: ***coeff_p1*** *0 1.0;1 10.0;2 20.0*	success
get	*expected_polynomial*	Compares the *current_polynomial* with the *expected_polynomial.*	success: if polynomials are identical failure: if polynomials are not identical
eval	*x expected_value*	Calculates the value of *current_polynomial* at *x* and then compares it with the *expected_value.*	success: if values are equal failure: if values are not equal

- **Adding and Multiplying Polynomials**
  Testing functions for addition and multiplication requires the definition of two polynomials prior to performing these operations. We use command "***coeff_p2***" to define the second polynomial.

Command	Parameters	Explanation	Output
coeff_p2	*second_polynomial*	Defines the second polynomial for an addition or a multiplication operation.	success
add	*expected_polynomial*	Adds the parameters of the ***coeff_p1*** to the parameters of the ***coeff_p2***, then compares result with *expected_polynomial*.	success: if polynomials are identical failure: if polynomials are not identical
mult	*expected_polynomial*	Multiplies the parameters of the ***coeff_p1*** with the parameters of ***coeff_p2***, then compares result with *expected_polynomial*.	success: if polynomials are identical failure: if polynomials are not identical

- **Test Files**

The course web site contains example input files with the corresponding output. The files are named *test01.in*, *test02.in* and so on with their corresponding output files named *test01.out*, *test02.out* and so on.

4. How to Submit Your Program

Once you have completed your solution and tested it comprehensively in your computer or on the lab computers, you have to transfer your files to the *ecelinux server* and test there since the automated testing is done using this environment. Once you finish testing in the *ecelinux server*, you will create a compressed file (tar.gz) that should contain:

- A document (maximum two pages) describing your design. This document must be typed for marking and should be submitted in PDF (*e.g.,* design.pdf).
- A test program (polynomialtest.cpp*)* containing your *main()* function.
- Required header files and classes (ending in *.h .cpp*).

The name of your file should be ***xxxxxxxx***_p***n***.tar.gz, where ***xxxxxxxx*** is your UW user id (*e.g.*, jsmith) and ***n*** is the project number that is o (zero) for this submission.