Lab 4

Due Mar 25 by 8am **Points** 10 **Submitting** a file upload **File Types** pdf and tgz
Available until Mar 25 at 8am

This assignment was locked Mar 25 at 8am.

For this lab we are using a linked list to store polynomials.

•• P14.5

Write a class `Polynomial` that stores a polynomial such as

$$p(x) = 5x^{10} + 9x^7 - x - 10$$

as a linked list of terms. A term contains the coefficient and the power of x. For example, you would store $p(x)$ as

$$(5,10),(9,7),(-1,1),(-10,0)$$

Supply member functions to add, multiply, and print polynomials. Supply a constructor that makes a polynomial from a single term. For example, the polynomial `p` can be constructed as

```
Polynomial p(Term(-10, 0));
p.add(Polynomial(Term(-1, 1)));
p.add(Polynomial(Term(9, 7)));
p.add(Polynomial(Term(5, 10)));
```

Then compute $p(x) \times p(x)$.

```
Polynomial q = p.multiply(p);
q.print();
```

Notice from the description above that Term is also a class. It contains the coefficient and exponent of one term plus a pointer to the next node of the linked list. The Polynomial class then encapsulates the list itself by using a Term pointer as data member to represent the *head* of the list. You may also use a *tail* pointer if it is useful.

Our GUI program allows the user to enter one term at-a-time in any order and stores them into the Polynomial using an *"insert in order"* member function. The order is by exponent and could be increasing or decreasing as you choose. Use Cairo to display the 2 polynomials the user enters as well as the sum and product in nice math notation (i.e. subscripts using **unicode (https://en.wikipedia.org/wiki/Unicode_subscripts_and_superscripts)** is easiest way).

Here are some samples:

Addition (http://www.mesacc.edu/~scotz47781/mat120/notes/polynomials/add_subtract/add_subtract.html)
Multiplication
(http://www.mesacc.edu/~scotz47781/mat120/notes/polynomials/multiplying/multiplying_poly.html)

Use ***Catch2*** *to* unit test your **add** and **multiply** Polynomial class *member functions*, then incorporate those into the **FLTK/Cairo** GUI we have developed and capture interactions with the user as images to include in your pdf file.

Add comments on each page with brief explanations of the important design decisions you make.

Use **cpp2pdf** to create the PDF that will have all source, code, images and comments nicely formatted and easy to read. Use the Latex **\newpage** command where needed to make logical breaks in the program structure.

Save your work by using this command in the working directory (lab4), retrieve the tgz and pdf file using the web server, and upload and submit to Canvas.

```
save . *.pdf
```