

CMPS 148 – Homework #6

100 Points

Design and implement a class for general polynomials. The polynomial

$$a_n X^n + a_{n-1} X^{n-1} + a_{n-2} X^{n-2} + \dots + a_0$$

Your class will be implemented as a **linked list**. Each **node** will contain an **int** value for the power of x and an **int** value for the corresponding coefficient. The operations should include **addition** and **evaluation**.

Important: You can adapt the linked list we developed in class, but you may not use STL (vector) for this assignment.

Evaluation of a polynomial is implemented as a member function with one argument of type **int**. The evaluation member function returns the value obtained by plugging in its argument for x and performing the indicated operations.

The class should include a method for adding/modifying a term, called **SetTerm** that accepts the power of the new term and its coefficient (both of type **int**). If a term with the same power already exists, simply overwrite its coefficient with the new one.

The class must also include a method for retrieving the coefficient for a given term. The operations **GetCoefficient** should be defined. This method should accept the power (**int**) to be retrieved and return the coefficient corresponding to that power. *Note, if there is no term with the given power, then return 0.*

Finally, the class must be able to print itself. Implement a print function that prints the polynomial above using \wedge symbols to represent the power part of each term.

Example: $6x^2 + 5x - 5$ should print out as $6x^2 + 5x - 5$

Submit a test program that creates multiple polynomials and performs all the supported operations. This test program can involve user input, or it can simply automatically generate its polynomials (perhaps using the `rand()` function). Your test program should prove to the user running it that all operations are indeed working correctly.

Examples:

The user of your class should be able to create an instance of your class and populate its terms as follows:

```
Poly test;  
test.SetTerm(1, -5);  
test.SetTerm(3, 2);  
test.SetTerm(0, 9);
```

The above lines of code create the polynomial: $2x^3 - 5x + 9$

To evaluate, the user of the code can call the Evaluate method with a given integer:

```
int result = test.Evaluate(2);
```

After executed, the result variable should be equal to $2 * 2^3 - 5*2 + 9 = 15$

Similarly, creating a second instance and adding it to the first should be accomplished by the following lines of code:

```
Poly test2;  
test2.SetTerm(1, 1);  
test2.SetTerm(2, 7);  
test.Add(test2);
```

After executing the lines above, test should be the polynomial: $2x^3 + 7x^2 - 4x + 9$

*Note, feel free to overload the + - * operators as well.*

Note: When printing, you should always print the in descending order with respect to the power of each term. It should not matter which order the SetTerm method are called! Your print function should never print terms with a coefficient of 0!