CPSC 2430 Data Structures

Homework Assignments #1

Assigned: 09/26/2016

Due by 9:20AM, 10/03/2016, Monday

1. Problem

You need to design and implement a polynomial class named Poly that represents a polynomial of a single variable ***x*** with degree ***n***: $a_0 + a_1x + a_2x^2 + \ldots + a_nx^n$, where $a_0, a_1, \ldots, a_n$ are integer constants called the **coefficients** of the polynomial. In this assignment, we assume $n \le 5$. Please refer to https://en.wikipedia.org/wiki/Polynomial for more details on polynomial.

For example, **$1 + 3x - 7x^3 + 5x^4$** is a polynomial of degree 4 with integer coefficients 1, 3, 0, -7, and 5. A common implementation of a polynomial stores the degree of the polynomial and the list of coefficients.

The Poly class must support the following operations (**do not change the class name and function prototypes**) as shown in the table below.

`Poly()`	Constructor: to construct an empty polynomial
`void read(istream& in)`	Read the polynomial degree and coefficients from *in*. For example, if we want to input a polynomial $1 + 3x - 7x^3 + 5x^4$, it shall first read degree: 4. Then, it shall read coefficients 1, 3, 0, -7, and 5 in the order. In other words, the first non-negative number shall be degree, followed by a list of coefficients. Given a list of integer inputs 4, 1, 3, 0, 7, 5, the polynomial is $1 + 3x - 7x^3 + 5x^4$.
`void write(ostream& out) const`	Write the polynomial to *out*. For a polynomial $1 + 3x - 7x^3 + 5x^4$, it shall be written to *out* in the form of 1 + 3x – 7x^3 + 5x^4. The terms with a coefficient of zero shall not be written.
`Poly operator+(const Poly& p)`	Overloading operator +.
`Poly operator-(const Poly& p)`	Overloading operator -.
`int evaluate(int v)`	Return the value of the polynomial for the variable x = v. The argument v should not be too big. Otherwise it could cause integer overflow.

Question: what data structure is good to store the coefficients of a polynomial?

2. Submission

You need to submit the following files:

- poly.h: header file for Poly class. You need to include Redundant Declaration in the header file.
- poly.cpp: implementation file for Poly calss.
- client.cpp: a client program to test the Poly class
- Makefile (you can reuse the Makefile in Lab1 by some minor revisions)

In your client.cpp, you should have some code like below to test your Poly class:

```
Poly p1, p2;
p1.read(cin);
p1.write(cout);
p2.read(cin);
p2.write(cout);
Poly p3 = p1 + p2;
p3.write(cout);
Poly p4 = p1 - p2;
p4.write(cout);
```

Before submission, you should ensure your program has been compiled and tested (extensively). Your assignment receives zero if your code cannot be compiled and executed.

You can submit your program multiple times before the deadline. The last submission will be used for grading.

To submit your assignment, you should follow two steps below (assuming your files are on cs1.seattleu.edu):

1). Wrap all your files into a package, named ***hw1.tar***

```
tar -cvf hw1.tar poly.h poly.cpp client.cpp Makefile
```

2). Submit your newly generated package ***hw1.tar*** as the first programming assignment ***p1***

```
/home/fac/zhuy/class/CPSC2430/submit p1 hw1.tar
```

3. Grading Criteria

Label	Notes
1a. Submission (1 pt)	All required files are submitted.
1b. Compilability (2 pts)	Your Makefile can compile the code and generate the executable file.
1c. Format & Style (1 pt)	Clean, well-commented code. No messy output/debugging messages. For C++ comments, please refer to https://google.github.io/styleguide/cppguide.html#Comments for more details.
1d. Functionality (6 pts)	The Poly class should behave as specified. All five operations are implemented and tested.
1e. Overriding policy	If the code cannot be compiled or executed (segmentation faults, for instance), it results in zero point.