Operator Overloading (OOPS)

Q 1) Create a class fraction to represent a fractional number and extend it to include all of the following operators:
+,-,*,/, <, >,<=,>=, !=, ==,(unary)-,++(prefix and postfix), --(prefix and postfix). Make +,-,*,/, <, >,<=,>=, !=, == friend functions

Note: The unary minus is a member function, which just returns the negation of the fraction that calls it: *Fraction operator-() const;*

Make the prefix operator a member function not a friend:*Fraction operator++();*

Include new member functions *getNumerator()* and *getDenominator()* which return the numerator (an int not a fraction!) and denominator of a fraction. Also write a program which sorts fractions using bubble sort .Use a suitable client code to test your class.

Q 2) Develop a class polynomial. The internal representation of a polynomial is an array of terms. Each term contains a coefficient and an exponent. The term $2x^4$ has a coefficient of 2 and an exponent of 4. Develop a full class containing proper constructors and destructor functions as well as set and get functions. The class should provide the following overloaded operator capabilities:

- Overload the addition operator (+) to add two polynomial
- Overload the subtraction operator (-) to subtract two polynomial
- Overload the multiplication operator (*) to multiply two polynomial
- Overload the assignment operator to assign one polynomial to another
- Overload the addition assignment operator (+=) , the subtraction assignment operator (-=), and the multiplication assignment operator (*=)

Q 3) The goal of this program is to implement a *set* of characters. For this assignment you should:

- Build a set class with functions *union, intersection* and *difference*.
- Supply a function which returns the number of elements in a set (*int getSize()*)
- Supply a function which makes a set empty. (*void erase()*)
- Supply a boolean function which determines whether or not a set is empty *(bool isEmpty()*)
- Supply a boolean function which determines whether or not one set is a subset of another (*overload operator<=*)
- Supply a boolean function which determines whether or not a given character is a member of a set. (*bool isMember(char c*)
- Overload the + operator for union, the * operator for intersection, the - operator for set difference, the <= operator for subsets and the == operator.

The + operator should also be overloaded so that if A is a set the expression A = A + 'e' adds the character 'e' to A. Thus the + operator can be used as A+B where A and B are sets and also A+'x' where A is a set and 'x' is a character.

Note:

- If A and B are sets A-B is the set of all elements in A which are not in B.
- If A and B are sets A*B is the set of all elements in BOTH A and B
- If A and B are sets A+B is the set of all elements that are in A and/or B
- A set never has duplicate elements. So if A = {'a','b','c} and b = {'a','b'} then A U B = {'a','b','c'}.

Q 4) Define two classes 'Polar' and 'Rectangle' to represent a co-ordinate point in the polar and rectangular systems. In polar system we represent a point as (radius, angle) and in rectangular as (x, y). Use convention constructor to convert from one system to the other

Q5) Stack class with
- → cout << s1 (overload)
- → $S_1 + S_2$ (append stack)
- → cin>> s1 (overload)