COEN 244 (Summer 2019) - Assignment 5

Deadline: Aug 18, 2019 by 11:55PM (strict)

Type: Group Assignment

Submission: Electronic Submission through your Moodle course homepage. Create an appropriate zip archive including all the C++ source files as well as your answers for the problem analysis and design steps.

Polynomial Calculator

Design and implement a software program that can perform algebraic operations on polynomials and rational functions. The software mainly consists of classes "Polynomial" and "Rational" which are described in the following:

"Polynomial" class:

This class manages and performs operations on polynomials of the following form: $a_n x^n + a_{n-1} x^{n-1} + \cdots + a_2 x^2 + a_1 x + a_0$, where n is a non-negative integer and a_0, a_1,a_2, ..., a_n are constant coefficients of type double.

The class offers the following, publicly available, services:

- **Constructor:** The constructor `Polynomial (string)` accepts a string representing the polynomial as input. If the string is well-formed the constructor will dynamically allocate and initialize the required memory to store the polynomial. If the string is not well-formed an exception shall be thrown. As an example of a well-formed string, the polynomial

  $4.5x^8 + 3.4x^3 - 6.4x + 3.1$

  should be string-encoded as follows:

  "4.5x^8+ 3.4x^3-6.4x^1+3.1x^0"

- **The Big Three:** Customized implementations of:
  - **copy constructor:** `Polynomial (const Polynomial &)`
  - **assignment operator**: `Polynomial& operator= (const Polynomial &)`
  - **destructor**: `~Polynomial()`

- **Polynomial Addition and Subtraction:**
  - Polynomial Addition:
    `const Polynomial operator+ (const Polynomial&, const Polynomial&)`
  - Polynomial Subtraction:
    `const Polynomial operator- (const Polynomial&, const Polynomial&)`

  Polynomial addition and polynomial subtraction is defined as the pairwise addition/subtraction of the respective coefficients. Consider the following example for polynomial addition:

$$(3.2x^5 + 1.6x^2 - 3.1) + (-1.3x^4 + 3.8x^2 - 4.3x) = 3.2x^5 - 1.3x^4 + 5.4x^2 - 4.3x - 3.1$$

- **Scalar Polynomial Multiplication:**
  `const Polynomial operator* (const Polynomial& P, double z)`

  The scalar multiplication $P * z$ simply multiplies each coefficient in P with z as depicted in the following example: $0.5 * (3.2x^5 + 1.6x^2 - 3) = 1.6x^5 + 0.8x^2 - 1.5$

- **Input / Output Operators:**
  - Input operator: `istream& operator>> (istream &, Polynomial&)`
  - Output operator: `ostream& operator<< (ostream &, const Polynomial&)`

  >> reads in a string representing the polynomial. If the string is mal-formed an exception should be thrown.

  << prints out the polynomial string.

- **Direct Value Access:**
  - Subscript operator (l-value): `double& operator[] (int n)`
  - Subscript operator (r-value): `double operator[] (int n) const`

  `p[n]` returns the value of the n'th coefficient of the polynomial p either as l-value or as r-value. If n does not denote a valid index (i.e., $n < 0$) an exception shall be thrown.

- **Function Operator:**
  - Function operator: `double operator() (double x) const`

  The function operator evaluates the polynomial at a certain value of x. E.g., consider the following polynomial p: $2.1x^2 + 0.8x - 1.5$. Then $p(1.2) = 2.484$ or $p(0) = -1.5$

"Rational" class:

The class represents rational functions as ratios of two polynomials.

$$y = \frac{a_n x^n + a_{n-1} x^{n-1} + \ldots + a_2 x^2 + a_1 x + a_0}{b_m x^m + b_{m-1} x^{m-1} + \ldots + b_2 x^2 + b_1 x + b_0}$$

The class offers the following, publicly available, services:

- **Constructors:** Objects of class "Rational" can be created by either passing on strings representing the polynomials or references to objects of class "Polynomial".

```
Rational (string, string)
Rational (string, Polynomial&)
Rational (Polynomial&, string)
Rational (Polynomial&, Polynomial&)
```

  If the object cannot be created because of a mal-formed polynomial string, an exception shall be thrown.

- **Output Operator:**
    - Output operator: `ostream& operator<< (ostream &, const Rational&)`

    << prints out the rational function as a fraction of its two polynomials.

- **Function Operator:**
    - Function operator: `double operator() (double x) const`

    Similarly to the function operator for the “Polynomial” class, this function operator evaluates the rational function at a certain value of x. An exception shall be thrown, if the denominator evaluates to 0.

Question:

a) Define and implement the `Polynomial` class and `Rational` class with all before-mentioned services and characteristics.
b) Write the driver program such that each member function, constructors / destructor, and overloaded operator is invoked at least ones.