**Assessment item 2**

**Assignment 2**

**Value:**20%

**Due date:**14-Apr-2014

**Return date:**05-May-2014

**Submission method options**

EASTS (online)

**Task**

Assessment 2 has total **40** marks. Marks will be scaled according to the value of the assignment.

Linked lists are a convenient way to represent polynomials. Your task is to design a program that does some polynomial operations using a linked list.

A polynomial of degree 𝑚 is defined as,

*a0xm+a1xm−1+...+ am−1x1+am*

The degree 𝑚 is the largest exponent in any of the terms of the polynomial. For example, the largest exponent in 4x2−7x+3 is 2.

It is customary to write the term with the largest exponent first, followed by the term with the next largest exponent and so on.

ais are the coefficients of the polynomial. One example of a polynomial of degree 6 is

2x6+x3+5x+4

in which the first term's coefficient is a0=2 and its power is 6; the second term's coefficient is a1=1 and its power is 3.

Tasks are detailed as follows

* You are required to develop your own class to represent polynomials as a linked list. The first node in the list represents the first term in the polynomial, the second node represents the second term, and so forth.
* Each node contains three fields. The first field is the term's coefficient. The second field is the term's power, and the third field is a pointer to the next term.
* The following examples show how a polynomial is represented as a linked list.

5x4+6x3+7

2x3−7x2+3x

Notice that in the linked list:

1.      The list has as many nodes as there are terms in the polynomial.

2.      Each node in the linked list contains a value for the coefficient, a value for the exponent and a reference.

3.      The nodes in the linked list are ordered. Nodes with a higher exponent precede nodes with a lower exponent.

4.      The constant term has an exponent of 0.

5.      The reference of the last node is NULL.

**What to Implement (Task):**

* You should have at least **two constructor** for polynomials. One is used to **build a polynomial (linked list) from a file** and the other from **keyboard input (pairs of coefficients and powers) by a user**.
* Implement polynomial addition by using an *add()*method, so you could write in program :

*PolynomialThree= PolynomialOne.add(PolynomialTwo);*

The rules for the addition of polynomials are as follows:

1. If the powers are equal, the coefficients are algebraically added.

2. If the powers are unequal, the term with the higher power is inserted in the new polynomial.

3. If the exponent is 0, it represents x0, which is 1. The value of the term is therefore the value of the coefficient.

4. If the result of adding the coefficients results in 0, the term is dropped (0 times anything is 0).

For example, the polynomial sum (27x3+15x2+12x)+(18x2+10x+8)is calculated as

27x3+33x2+22x+8

* Get the value of a polynomial given an input. For instance, given x=2, the value of the polynomial x2+3x+2 is 12 .
* Implement the product of a polynomial by another polynomial which is the algebraic sum of the products of one polynomial by all the monomials of another polynomial.
* Implement the derivative operation for a polynomial. Note that for a monomial axk, its derivative is kaxk -1and the derivative of a constant is 0 and the derivative of a polynomial is equal to the sum of derivatives of its all monomials.
* Implement a *InsertTerm()*method. This method should put the input term into a polynomial.
* Test your program with the following two polynomials

12x9+5x6+13x5−4x4+2x3+11x2- 6

7x9+2x8+5x6+2x5+2x3+9x2- 7x

* Write a driver program to test all the functionalities that you have defined

**Rationale**

This assessment is designed to:

1. test students' ability in programming with Java

2. Practice their knowledge in fundamental data structures

**Marking criteria**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Assessment 2 (Total marks 40)** | | | | | |
| Criteria | Marks | Pass | CR | DI | HD |
| **a.    Execution** | 4 | Programs execute without trouble (4). 0.5 mark will be deducted for each program that fails. | | | |
| **b.    Design & implementation** | 25 | Two constructors (4), polynomial addition (3), polynomial value (3), product of polynomials (3), derivative operation for a polynomial (3), InsertTerm method (3), Test results (3), driver program (3). To get "Pass" coding with minor errors required and to get "HD" all codes must be correct, professional with appropriate comments. | | | |
| **c.     Presentation** | 3 | Few comments and minor indentation errors (1.5) | Proper indentation in each code block and comments in major lines and blocks (3) | | |
| **d.    Submission** | 8 | Provide necessary files (4), Discussion on algorithm, instruction on compiling and running your program (4). | | | |

**Presentation**

o programming solutions to the tasks;  
o javadoc documentation of your classes;  
o any input files required to run your programs and examples of output files; and  
o a discussion of important algorithms used and how your programs should be compiled and run.

Please submit your project through EASTS system.