**Polynom\_able**

**Assigned Authors:** Teddy Grossman, Orad Cohen.

**Programing language used:** Java.

**Assigned Work:** Polynom class, Monom class, PolynomTest class, MonomTest class.

**Packages:** myMath.

**Classes in Package:** cont\_function, function, Monom, Monom\_Comperator, MonomTest, Polynom, Polynom\_able, PolynomTest.

**Short Description**

This Interface represents a polynomial of the form , Our implementation of Polynom\_able is done via the class Monom which represent a single monomial of the form , and Polynom, which uses HashMap data structure to store and handle. Our implantation allows the following functionalities as specified below.

**Design decisions and issues.**

We have Implemented Hash map as follows: The Polynom class store each Monom object in hash map using the monom’s exponent as key. Using HashMap gives us lookup advantage as opposed to other structures like linked list.

We encountered a problem when using toString(), we found it hard to sort the monoms by their exponent. We solved it using Map and TreeMap library to sort the hashmap. But since some of our functions relay on toString, we partially sacrificed our lookup speed for aesthetic.

Additional problem is the double precision in java, which we solved by cutting the double after the 6th figure after the decimal point.

We should have planned in advance instead of diving in, caused a lot of compatibility problems between our methods which took time to fix. We will definitely learned from this assignment.

**Testing:**

The following assignment has been given by us a pre-made tester filled with an array of Strings which are categorized into 2 groups, good and bad.

The good has Polynom Strings that should work as intended, while the bad purposefully has Polynom Strings that should return an "Error" and tests our problem catching.

You may add additional Strings to each array if you choose, just keep in mind that the correct format of a Polynom String should be in the form of: ""

While each is a Monom.

**Polynom Class Features**

**Constructors:**

Polynom() – creates a zero polynom.

Polynom(String s) – allows you to initialize a polynom using a string. (E.G – Polynom Poly = new Polynom(“5x^2-3+4x^12”).

**Methods:**

Polynom\_able copy()- Return a deep copy of the Polynom.

Add(Polynomal/Monom) – add a monom or a polynom to the Polynom.

Subtract(Polynoml) – subtract a Polynom of the Polynom, and remove a monom of the HashMap if the coefficient is 0;

Multiply(Polynomal/Monom) – multiply the polynom by a monom or a polynom.  
Equals(Polynomial) – return true if given polynom equals to the polynom.

isZero() – return true if the Polynom is empty/has 0 as coefficient

Derivative()- return a derivative of the polynom

f(double x)- returns a double of the value from the Polynom function at given x.

toString()- return a string of the polynom from HashMap (e.g. “5x^2-3+4x^127”).

Root()- Uses a sort of binary search to return the middle f(x) point between two points on the function (x0,x1) assuming that x0 \* x1 <= 0.

derivative()- returns a Polynom\_able that is the derivative of the original Polynom.

Area()- returns s double which is the positive f(x) area of a given function using Riemann's Integral.

Iterator()- function to iterate through a Polynom's Monoms.

Sortbykey()-returns a String of a Polynom but at a descending order.

**Monom Class**

**Constructor:**

Monom(string s)- Initialize a Monom using strings. (E.G. Monom mon = New Monom(“4x^2”);.

Monom(double a,int b)- Initialize a Monom with double as coefficient and int as power of x.

Monom getNewZeroMonom()-Initialize a 0 Monom.

**Methods:**

Get/Set coefficient/Power() – self explanatory.

Add(monom x)- add the coefficient of x to the monom.

Multiply(monom x)- multiply the coefficients, and add their power.

toString()- return a Monom String in the form a1x^b1 (e.g. “4x^2”).

Monom derivative()-return a derivative of the Monom.

f(double)- returns a double of the value from the Monom function at given x.

isZero()-return true if the Monom is empty/has 0 as coefficient

roundAvoid()-round a double by cutting it after the 7th figure after decimal point

**For Further Work on the Project:**

Keep in mind that the classes worked on are implementations(subclasses) of various super classes. Please consult the lead programmer before doing any drastic changes to the super classes.

**License:** All rights reserved to Ariel CS department