

## HW 1

*Applied Software Engineering for Engineers and Scientists*

EE/CIS 694T

Spring 2000

A scientific application needs to manipulate **polynomials**. For this homework, restrict yourself to **zeroth** order ( $a_0$ ), **first** order ( $a_0+a_1*x$ ), and **second** order ( $a_0+a_1*x+a_2*x*x$ ) polynomials with **real coefficients** ( i.e.  $a_0$ ,  $a_1$ , and  $a_2$  are real numbers). The application needs **real roots** of these polynomials, as well as **real solutions** of **equations** of the form,  $polynomial1 = polynomial2$ .

Think about **abstractions** of **polynomials** and **equations** in this problem, i.e. treat polynomials and equations as **objects**. Think about what **interfaces** and **classes** you would need for this application. Keep **flexibility**, and **extensibility** in your mind while designing a solution. i.e., in the future the application might need to manipulate higher order polynomials.

*Think carefully about issues like:* How are the concrete classes in your design instantiated? Are your objects being instantiated with a valid initial state? i.e., if you are creating a second order polynomial object (abstraction of,  $a_0+a_1*x+a_2*x*x$ ), then how do you ensure that its coefficient,  $a_2$  is **non zero**? Otherwise it would be a first order polynomial. How do you handle the situation when a user of your classes tries to instantiate a second order polynomial object with  $a_2$  equal to zero? Is code being duplicated in various classes? How can you make different classes reuse code? Which class should be responsible for determining the roots of a quadratic equation? How are you **distributing the behavior** (functionality) of your application among your interfaces and classes? i.e., which methods belong to which interfaces or classes? Which methods should be **public**, and which should be **private**?

Implement your interfaces and classes in Java and place them in a package named **polynomials**. Place a class named **Client** in a package named **polyDemo**. Implement a **main** method in this class. This **main** method should **demonstrate** how a **client** would use your interfaces and classes in the package **polynomials**. In this method you should create **first order** and **second order** polynomial objects, and invoke the appropriate methods on them to find their **real roots**. This method should also demonstrate the creation of an **equation object** (abstraction of  $polynomial1 = polynomial2$ ) and the invocation of the appropriate method on this object to determine its **real solutions**.

Your homework should consist of your **source code** (along with a full **output listing**), and a simple UML **static class diagram** of your design. The class diagram should indicate all the **attributes** and **operations**, as well as the **associations** and other relationships like “**implements**” and “**extends**” (if these relationships are present in your design). Also include a short description of the design decisions you have made to make the resulting code more **flexible**, and **extensible**.