## Homework 4 – Part I: Using XML with DOM and Schema validation

**Due: Dec. 15, 2004 (no extensions)**

In this assignment, you will use XML, Schema definition and a C# or Java program to read in a data file including description of university employees. The data read from the input file should be used to create instances of "Person" objects to be stored in the University Directory. These are the details:

1- The input file is an XML instance file of a schema that you define.

2- The XML schema should define your rules for the "Person" related data to be entered. For example, you define that the "Faculty" element has the attribute "type" to indicate whether the faculty is an assistant or associate professor. You do not have to follow the same rules from the last assignment. You just need to have:
   a. At least two subclasses of "Person"
   b. At lest one of the subclasses should have an attribute to be specified in the XML file.
      i. Ex: <FACULTY Type = "Assistant">
   c. Each subclass can have any number of "subnodes", e.g., a Student can have a "Name" and a "GPA" tags:
      i. <STUDENT>
            <NAME> So AND SO </NAME>
            <GPA> 4.0 </GPA>
         </STUDENT>
   d. Please note the difference between *attribute* in the XML file, and *attribute* of an object in general. For example, in the examples above, "Type" is an XML attribute AND the attribute of the *faculty* object; whereas "name" and "gpa" are attributes of the *student* object, but are NOT XML attributes.

3- Once you read the input XML instance file using DOM, you will traverse the tree (as in *readingbooklist-w-dom.cs*), and create instances of *Person* objects and set their attributes based on the data read from the tree (as in *readingbooklist-w-schema,* where data is read into *book* objects). Each instance of *Person* is then stored in the University List, e.g., a Vector. So, for example, if you read a <STUDENT> node, you know you will have to instantiate a student as follows:
      i. Person p = new Student(*arguments*).

   The *arguments* above will depend on what the child nodes of <STUDENT> are

Thus, before you actually instantiate a *Student* class, you will have to read the child nodes, e.g., <NAME> and <GPA>, store their values in, for example, a *name* and *gpa* variables, and then create the intance of student: Person p = new Student(name, gpa). Now you only need to add the student to the list, e.g., studentlist.Add(p), where studentlist is an "ArrayList" (equivalent to Java's *Vector*).

4- You can use Java or C# for this assignment.

**Deliverable**

1- **An XML schema file**
2- **An XML instance file for the schema above**
3- **A well documented Java or C# source file.**
4- **A description of the rules you chose for your schema (which can also be embedded in the schema file as comments)**
5- **A short "user manual" describing how to use your program**
6- **No sample runs, please..**