Comp 212 - Intermediate Programming EXAM #1 October 03, 2001
Rice University - Instructors: Nguyen & Wong NAME: _____

Instructions

1. This exam is conducted under the Rice Honor Code. It is a closed-notes, closed-book exam.
2. Fill in your name on every page of the exam.
3. If you forget the name of a Java class or method, make up a name for it and write a *brief* explanation in the margin.
4. You are expected to know the syntax of defining a class with appropriate fields, methods, and inheritance hierarchy. You will not be penalized on trivial syntax errors, such as missing curly braces, missing semi-colons, etc, but do try to write Java code as syntactically correct as possible. We are more interested in your ability to show us that you understand the concepts than your memorization of syntax! If you don't remember the code, write your thoughts down in prose.
5. Write your code in the most object-oriented way possible, that is, with the fewest number of control statements and no checking of the states and class types of the objects involved.
6. In all of the questions, feel free to write additional helper methods or visitors to get the job done.
7. Make sure you use the Singleton pattern whenever appropriate. Unless specified otherwise, you do not need to write any code for it. Just write "singleton pattern" as a comment.
8. For each algorithm you are asked to write, 90% of the grade will be for correctness, and 10% will be for efficiency and code clarity.
9. You can reference the following visitors from the lectures and the homeworks: GetLength, GetMin, GetSum, ToString, Reverse, MakeClone, LastElement, GetNth, FirstNElements, ConcatWith.
10. You have two hours and a half to complete the exam.

Please State and Sign your Pledge:

1a) 10	1b) 5	1c) 5	2a) 15	2b) 10	3a) 10	3b) 15	3c) 15	4a) 10	4b) 5	TOTAL 100

1. Design an object model for the following *immutable* data structure that represents a web page (20 pts total):

- A web page has a header, which is a string, and a body, which is a web document.
- A web document may be empty or non-empty. When it is empty, it contains nothing. When it is not empty, it may contain a string and web document, or it may contain a web page and a web document.

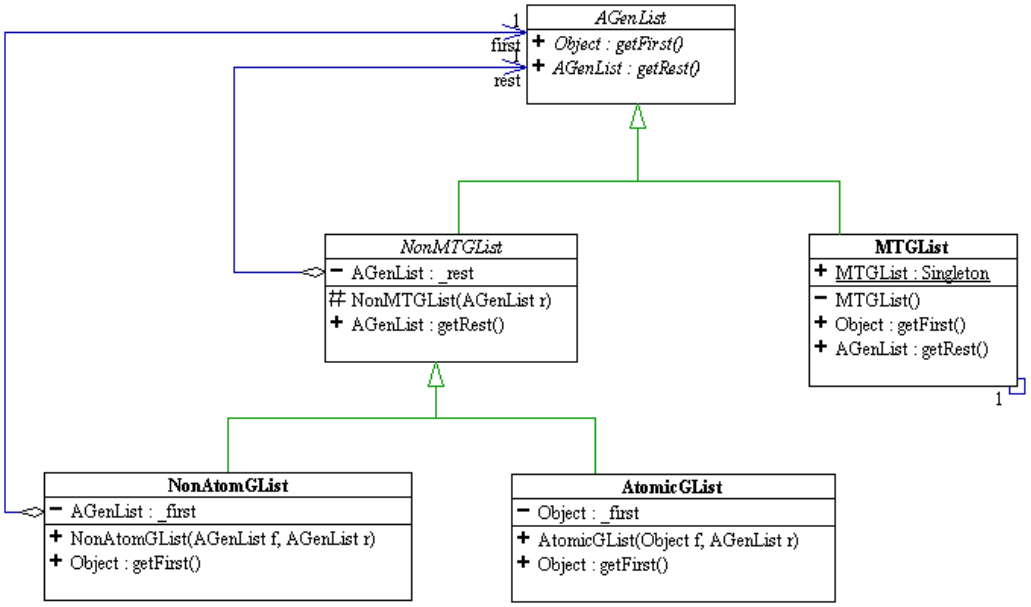
You will need to:

a) Draw the corresponding UML class diagrams. (10 pts)

b) Identify the design patterns in your design. (5 pts)

c) Justify the use of any abstract classes in your design. (5 pts)

2. Consider the following UML class diagram for generalized lists (lists that may contain lists), as seen in homework #2. (25 pts total)



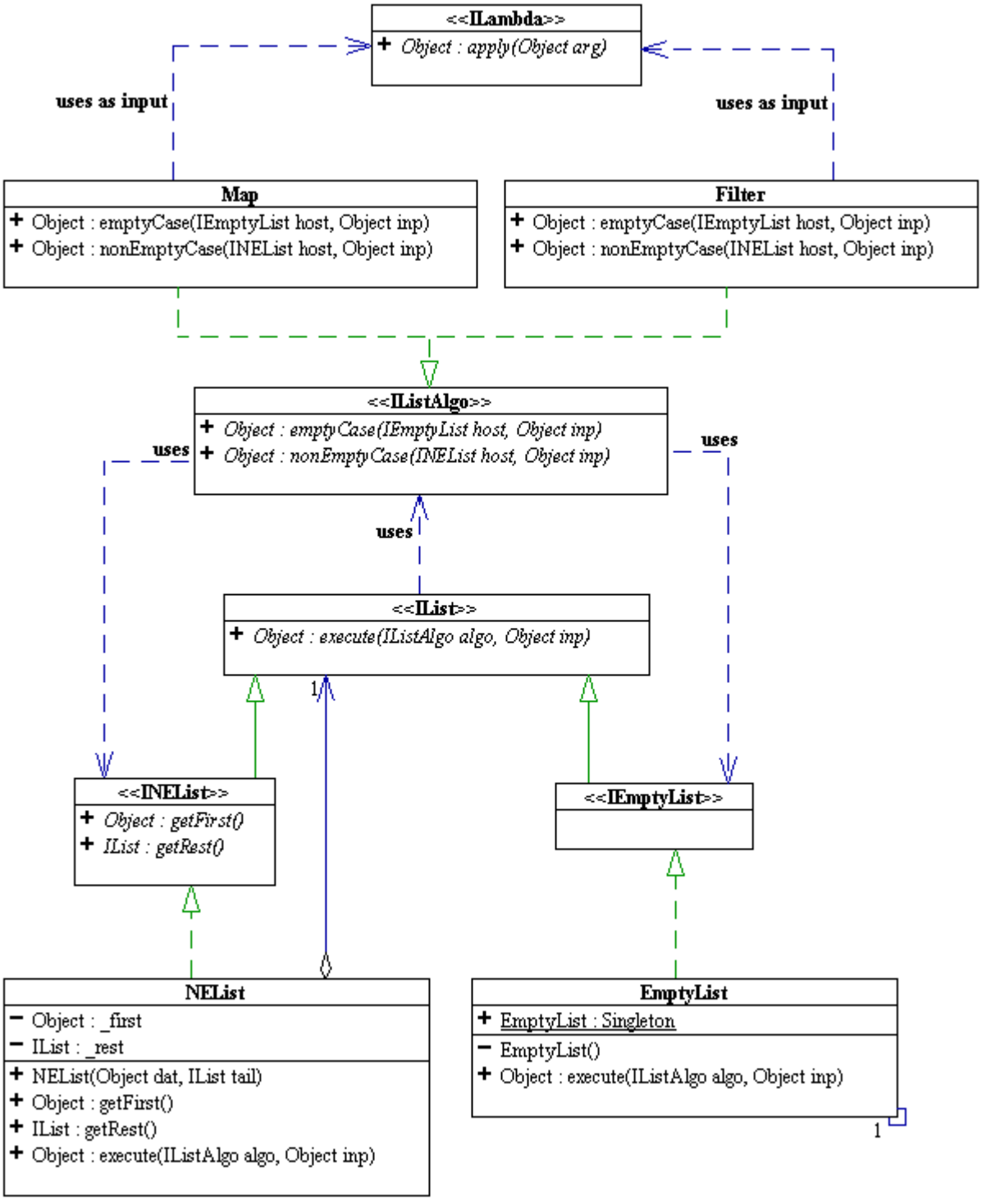
a) Suppose you are given an AGenList, gls, that contains String and/or lists of String. Add appropriate methods and write the code to locate a given String, w, and return an AtomicGList whose first is w (the search order is unimportant). If w cannot be found, then return null. Use the String method equals() for String equality. You do not have to write out the code for the whole class, just indicate to which class your methods belong. Write on the back of the page if you run out of space. (15 pts)

b) Explain what you need to do to apply the visitor pattern to the above structure. (10 pts)

Be specific as to exactly

- What classes and/or interfaces you need to add.
- What methods are added to what classes/interfaces.
- What the signatures of those methods are.

3. For this problem, you are to use the familiar immutable list framework, IList, given by the following UML diagrams. (40 pts)



In the above diagram the interface ILambda is a model of an abstract function that can take an Object as input and produce some Object as output via the *apply* method, which has an argument, arg.

a) Give an example of (i.e. write the code for) a concrete ILambda that takes Integer as input and returns a Boolean as output. Such an ILambda is called a *predicate* on the Integer. (10 pts)

b) Write a visitor called Map that takes as input an ILambda and returns an IList consisting of applying the ILambda to each element in the IList host. (15 pts)

c) Write a visitor called Filter that takes as input an ILambda predicate and when executed by an IList host, "h", returns an IList consisting only of elements in "h" for which the input ILambda predicate holds true. (15 pts)