1. Instructions

You will be continuing work with your group on this project. Your whole group will only need to submit one completed project. This can be done from any group members Blackboard account. Your solution must be uploaded to Blackboard by the deadline, read section 5 to learn what to submit and by when.

2. Project Objectives

The objective of this project is to have you become familiar with refactoring an object orientated program with inheritance and polymorphism on a team of engineers using the C++ programming language. Along with using the advanced features of the C++ language, my hope is that you become even more familiar with the importance of good design and testing your code.

3. Project Parts

This project will be composed of one part which you will turn in within two weeks. The project is composed of a design component, where you will need to create a UML class diagram which includes each class of your program. In this project you will need to turn in test cases for each method and test cases for the whole program as you did in lab 3. (25 points)

The second component will be; the coded implementation of your design and test cases. The code will need to be commented as well so that I can understand what each method is designed to do.

The code must compile for any credit more than 50% credit. You can use the provided project structure I have provided or code your own. You must have a make file for your implementation that compiles your code into an executable called “CrossReference”. This part of the project will be graded on how correctly the project runs. (25 points)

4. Problem Statement

You will need to refactor, or redo, some of the code you created in Lab 4 to introduce inheritance and polymorphism into the project. We want to expand the idea of a Token to a more refined definition so we need to make Token into a base class. Now our Binary Tree holds a special case of Token, specifically it holds an Identifier. Therefore our first expansion will be to create an Identifier Class, which is a child of Token, and modify our Binary Tree to only hold Identifiers. We then want to simplify the idea of a Literal. Specifically, we should get rid of LiteralType as a type, inheritance is much cleaner. Therefore we should make a child of Token that is a Literal Class. We also need to make children of Literal for all of the literal types. One class will be the Real class, another class will be a String class, and a Integer class.

Since Literal is never an object (only Real, String, Integers will be objects), we should make Literal into an abstract class with a pure virtual function. We can then make virtual functions that will be implemented in the children of Literal so that we can get and set our literals. The ultimate goal of this project is to remove LiteralType from the whole program. If you want to make your program even better you can make Literal into a Template class as well as an abstract class. If you make Literal a Template class it is worth 25 points extra credit on this assignment.

Since we want to specify the Identifier as its own type and put it into its own class, we can move the methods and data that we need for the Binary Tree into the Identifier class and clean up the Token class. Once this is complete we should change our Binary Tree to work with the Identifier class only.

A big challenge of this task will be to change the scanner class so that it creates the proper type of child object and then changes it to a Token object before returning it to main in the get token method. You should refer to your design documentation, test cases, and comments from previous labs to deduce what changes need to be made to the scanner class and then make them without breaking the whole program.

5. What to Submit for Grading and Assignment Deadline

The lab will be due Thursday April 28th at 11:59PM. You should include your design document(s) and the URL of your repository website so I can go to it and look at how everyone contributed. A portion of your grade will be determined by everyone contributing something to the repository. You will need to include a table clearly on your document that contains each team member's name. In that table you will place a number 0, 1, or 2 next to the name. Please see the class announcement I made on Blackboard for what this number system refers to. Failure to include the table with the team members names and a number next to their name results in all team members receiving 50% off for the project.

The documents must be in pdf format and submitted as one tar file. Please see project one for more information on this. You should include your commented source code, test code files, and your design document(s). The design includes UML, and must be in pdf format. Please put all files together as a tar file and provide a URL to your Repository website. You will need to include a list of each team members name and their corresponding repository ID. I will be going to your site and checking the actual contribution of each team member. If I see everyone is contributing then all team members will receive equal points. If I see some team members have contributed significantly less than the others I will further investigate and deduct a percentage of those team members points based on their contribution. If you are worried that one team member didn't contribute much to the repository, but they were instrumental in the project (for example the team leader), then you can comment on this when you submit the assignment. Please put your group members names at the top of every document you submit including code (this must be a comment). The file name of your tarball should be cse220_lab05.tar. Please see project one for more information on the penalty for not submitting a pdf file for documents that must be in pdf format.