**Project #14 – Program Animation**

**Purpose:**

This project provides a program animation facility that plots the execution trajectory of a program in a visual representation of the program’s scope tree. All code written in C, C++, C#, or Java have structure defined by a tree of scopes:

1. All code is contained in a nested set of namespaces with the top level being an anonymous global namespace.
2. Namespaces can contain other namespaces, functions, data declarations, classes, structs, and enumerations.
3. Functions can contain executable statements and declarations of data, classes, structs, and enumerations but can’t contain functions that don’t belong to a class.
4. Classes and structs can contain functions and data declarations.
5. Enumerations can contain only data declarations.

The purpose of this project is to trace the execution of a program by highlighting a series of nodes in its scope tree, based on its path of execution.

**Requirements:**

For the Program Animation Project you will:

1. Write code to analyze the scope tree using the Parser you will find here:   
   <http://www.lcs.syr.edu/faculty/fawcett/handouts/CoreTechnologies/Cpp/Code/Parser/>
2. Write code, using WPF, to display the scope tree in some convenient form (see UFP1 and UFP2).
3. Develop a logging class that writes its location and system time in an XML message sent to a message manager.
4. Develop a message manager that accepts the logger messages and stores them for later playback.
5. Develop an annotator program that inserts, at the beginning of each function in the code to be animated an instance of the logger class. The logger class constructor enqueues an enter message with the name of its function (passed as an argument) and the system time. The logger class destructor enqueues a leaving message with the system time.
6. When the annotated code is built and run the message manager saves the sequence of messages.
7. Develop a playback program that highlights nodes on the scope tree representation of 2 based on sequencing through the message set.
8. The playback program should provide for continuous show with a settable time interval between steps, single stepping and reversal.