CSCI 4160 Project1

Due: see class calendar

Goal:

This assignment serves several purposes:

- to be familiar with Visual Studio, the official IDE for this class;
- to practice on subversion control software and Tortoisegit client;
- to understand class inheritance and benefits of polymorphism
- to be familiar with symbol table, and abstract syntax tree

Description:

In this assignment, you are required to write an interpreter for a programming language (Don't be scared, you don't need to know the syntax and you never need to parse the program). The input to the interpreter is an abstract syntax tree of a program, which will be provided by the instructor. The output of the interpreter is the output of the program (i.e. interpreter it).

The language can have statements and expressions. Statements have no return value, while expressions have associated integer values.

The statements can be

- compound statements: a sequence of statements separated by semicolon.
- print statement: a statement to print the values of a sequence of expressions.
- assignment statement: assign an integer value to a variable

The expressions can be

- id expression: it is purely a variable
- integer literals like 10, 34.
- arithmetic expressions using operators +, -, *, /.
- Sequence expressions: a statement followed by an expression

We defined abstract base classes *Stm* and *Exp* to represent statements and expressions. Each type of statements/expressions is defined as an individual class derived from the abstract base class *Stm*/*Exp*. We also defined an abstract base class *ExpList* and two derived classes *PairExpList* and *LastExpList* to represent a list of expressions. All classes are defined in ***slp.h*** and you should implement all member functions in **slp.cpp**. What you need to do is to provide implementation to the following member function within each non-abstract base class:

1. void/int interp (SymbolTable&) that "interprets" a statement, an expression, or a list of expressions.

In addition to the function, you are also require to implement int size() member function in PairExpList and LastExpList, which should return the number of expressions in the list.

The symbol table is a hash map from string to integer. The symbol table is used to store variables and their values. So you should understand when to push a variable to the symbol table and when to lookup the symbol table. In this assignment, you never delete a variable from the symbol table..

The instructor provides the driver to test your interpreter on four different test cases which can be found in prog.h.

HOW TO GET STARTED

1. Set up your coding environment as described in the class notes document….see "how to prepare for projects".
2. Clone a local copy of the class repository. Please refer to Tortoisegit Tutorial on the course webpage. The folder projects/project1 in the class repository, contains the following skeleton files:
    - description1.pdf: this document
    - slp.h: definition of all classes
    - slp.cpp: skeleton file
    - prog.h: all four test cases
    - main.cpp: the driver
    - rubric1.doc: the rubric to grade your project.

    All your implementation should be placed in the file **slp.cpp**.
3. Create a .Net project using Visual Studio under project1 folder in your local repository. Save your entire .NET project. Copy all source programs to the folder of the project, then add them to the project in Visual Studio. After you finished your work, be sure to commit your files to local repository and push all changes to remote repository on ranger.
4. Once you have finished, submit the project in the following way:
    - Copy the file project/project1/rubric1.doc from the class repository to the project1 folder in your local repository. Edit the file to put your name.
    - Add rubric1.doc to local repository
    - Commit the whole project1 folder to your local repository.
    - Push latest version of project in your local repository to remote repository in ranger.
    - **Any commit of the project after the deadline is considered as cheating. If this happens, the latest version before the deadline will be graded, and you may receive up to 50 points deduction.**
5. You can check your grade by update the rubric1.doc from the repository after the notice from the instructor.