CS4850 L4850 Interpreter Project 2
Due Date: Tuesday, November 5, 2013 @ 8am.

Purpose

The purpose of this project is to gain experience in giving meaning to a programming language by interpreting expressions in L4850.

Project Summary

In this project, you will need to implement an interpreter of the abstract syntax that you generated in Project 1. For full credit, you must use the modified visitor pattern we developed in class. Spefically, your interpreter must be able to

1. Evaluate any abstract syntax generated for
   a) List, string, integer, floating-point and boolean constants,
   b) Arithmetic, relational and boolean expressions, and
   c) If expressions.
2. Perform dynamic type checking.
3. Evaluate function invocations for the L4850 language-defined function listed in Table 1 in the document "L4850: A Dynamically Typed Language for Instruction". You must catch any error that could occur in a language defined function and report a suitable error message. Specifically, no Java exception or error produced by performing a language defined function should be left uncaught.

Arithmetic and Relational Expressions

L4850 allows mixed type arithmetic expressions. For `+`, `-`, `*`, `/`, `<`, `<=`, `>`, `>=`, `==`, `!=`, type coercion rules are defined in Table tab:arith.

Left Operand Type	**Right Operand Type**				
	Float	Integer	Boolean	List	String
Integer	Float	Integer	*error*	*error*	*error*
Float	Float	Float	*error*	*error*	*error*
Boolean	*error*	*error*	*error*	*error*	*error*
List	*error*	*error*	*error*	*error*	*error*
String	*error*	*error*	*error*	*error*	*error*

Table 1: Arithmetic Type Coercion

Logic Expressions

Logic expressions are only defined on `booleans`. All other operand types result in a type error.

If Expressions

The type of the test for an if-expression must be a boolean. All other types are a type error. The type of the then- and else-expressions can be anything.

Solution to Project 1

I have provided my solution to Project 1 on E-Learning. You may download and use my source if you wish. In addition, I have provided a small set of test input on E-learning.

Requirement: Command-line

Your project is required to accept a file name on the command-line and read the input from that command line instead of the prompt. I have provided a solution to this in my code from Project 1.

Submission

Write all of your code in Java. Export your entire project to a zip file and submit that zip file via E-learning.