Compilers

Project Two

Write a complete lexer, parser, and type checker that validates the source code and creates a symbol table (including type and scopr data) for the following grammar.

```
Program         ::== Statement $

Statement       ::== print ( Expr )
                ::== Id = Expr
                ::== VarDecl
                ::== { StatementList }

StatementList   ::== Statement StatementList
                ::== ε

Expr            ::== IntExpr
                ::== StringExpr
                ::== Id

IntExpr         ::== digit op Expr
                ::== digit

StringExpr      ::== " CharList "

CharList        ::== Char CharList
                ::== Space CharList
                ::== ε

VarDecl         ::== Type Id

Type            ::== int | string

Id              ::== Char

Char            ::== a | b | c ... z

Space           ::== the space character

digit           ::== 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0

op              ::== + | -
```

Curly braces denote new scope.

Notes and Additional Requirements:

- Build a symbol table of `Id`s that includes their name, data type, scope, position in the source code, and anything else you think is necessary or important.
- Type-check the source code. Catch undeclared identifiers, redeclared identifiers in the same scope, type mismatches, and anything else that might go wrong. Issue warnings about declared but unused identifiers. Warn about use of uninitialized variables, but do not treat them as errors.
- Include verbose output functionality that traces the stages of the parser including the construction of the symbol table and type checking actions.
- When you detect an error report it in helpful detail including where it was found.
- Construct a syntax tree as you go. You'll need this for the next project.

alan@Labouseur.com
www.Labouseur.com

Create several test programs that cause as many different types of errors that you can think of in order to thoroughly test your code (remember "code coverage"). Include a few test cases that show it working as well. Write up your testing results in a document to hand in.

E-mail all of your files for this project to me before the beginning of the class in which this is due.

alan@Labouseur.com
www.Labouseur.com