CSCI 4160 Project5

Due: see class calendar

Description:

This is a team project that build on the previous project. In this project, you are required to implement a symbol table, and perform every basic semantic error checking listed below:

- undefined type/variable/function names
- redefined type/variable names.

All other semantic errors will not be covered in this project.

Classes in this project

Several classes are defined in different namespace for various purposes. Here is a short explanation of each class.

- Class absyn::Absyn and all its children are used to define types of nodes in the abstract syntax tree.
- Class template symbol::SymbolTable provides the implementation of symbol tables used in this course.
- Class types::Type and all its children are used to represent types supported by tiger languages. We are not going to use them in this project. But we need them to make sure the syntax is correct, and no change is needed for the next project.
- Structure symbol::SymTabEntry provides information that should be tied with a variable/function/type name in the SymbolTable. The SymTabEntry contains three information: the level of associated name in the source program, a pointer to the type information of the name, and a pointer to the AST node that contains the variable/function/type.
- Class symbol::Env provides the compile environment for Tiger language. It has two member data of symbol::SymbolTable<SymTabEntry>, one to store variable/function information, and one to store type information as specified by Tiger language manual. Its constructor will insert all global functions like print, ord into variable symbol table, and built-in data types like int, and string into type symbol table.
- Class semantics::TypeChecking provides functions to check semantic errors at each node in the abstract syntax tree..

Your major task is to complete the implementation of the following classes.

1. SymbolTable: this class define a class template for symbol table
2. TypeChecking: this class performs the basic type checking using the symbol table

You should not modify other classes in the project.

Tips:

1. Tiger.tab.hh, Tiger.tab.cc, lex.yy.cc files are provided by instructor in SymbolTableProject. So you don't need to use your own tiger.ll or tiger.yy files in this project. When you compile the solution, please just build the MainDriver project.

2. Every time an item is inserted into the variable or type symbol table, a SymTabEntry object should be construct. In this project, please pass NULL to the second parameter when constructing a SymTabObject. Let d be a pointer to an absyn::VarDec object, the following statement will insert it into the variable symbol table:

```
insertVar( d->getName(),
           symbol::SymTabEntry(
                env.getVarEnv()->getLevel(),
```

```
                NULL,
                d)
        );
```

where insertVar function is provided in the class semantics::TypeChecking

Test Report:

In this project, implementation of the project and test case design can be performed simultaneously. Don't wait too long to design test cases. In this project, please provide a tiger program called: test.tig to cover all of the following scenarios:

- Two variables with the same name are defined at the same level.
- Two types with the same name are defined at the same level.

- Two variable with the same name are defined at the different nesting levels.
- Two types with the same name are defined at the different nesting levels.

- An undefined variable is used
- An undefined type is used
- An undefined variable is used in function parameter list

Important Dates and Deadlines:

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
		14 (Oct.) **Project assigned**	**15**	**16** **1st time log report due**	**17**	**18**
19	**20**	**21**	**22**	**23** **Test.tig due** **Alpha version due** **2nd time log report due**	**24**	**25**
26 **Test Report due** **Modified Test.tig due** **3rd time log report**	**27**	**28** **Project Due**	**29**	**30**	**31**	

Notes:

- All reports/files are due at 11:30pm on the due date.
- Each team is required to submit 3 time log reports.

The following files are provided by the instructor in D2L dropbox:

- SymbolTableProject folder. This contains a sample Visual Studio 2010 project.
- Description5.pdf: this file

- Rubric5.doc: the rubric used to grade this assignment.
- example.txt. sample output
- Testreport.doc: format of testing report for this project

How to submit

Submit the following files to D2L dropbox before each dealine:

- 3 time log reports
- test case
- test report
- alpha version: alpha_TypeChecking.cpp and alpha_SymbolTable.cpp
- final version: TypeChecking.cpp and SymbolTable.cpp