CS322 – Language and Compiler Design II

Assignment 1 – Semantic Checking

Requirement

In this assignment, you will build a semantic-checker for the SimpleC (C--) program. To do this, you'll process declarations to extract type information about identifiers and process statements and expressions to make sure that identifiers and literals are used correctly. Your checker should detect all semantic errors and warnings, including:

- incompatible or illegal operand types in expressions, statements, or declarations;
- uses of undefined symbols;
- multiple declarations of a symbol within a single scope;
- attempting to assign to a constant;
- incorrect number or types of arguments in function call;
- `return` statements in inappropriate contexts, or lacking a return value when one is required or vice-versa;
- …

Do not attempt to check:

- functions always execute a `return`;
- array references are within bounds;
- …

Your checker must be implemented within file **`Ast.java`**, as new methods (and new classes if necessary) with the main method **`void check()`** in class **`Ast.Program`**.

If the checker encounters any semantic error or warning in the program, it should call the method **`Errors.semanticError`** or **`Errors.semanticWarn`** with the line number at which the error or warning occurs and a suitable explanatory message.

Implementation

The start project contains the skeleton that supports lexical and syntax analyzing. You are strongly urged (though not required) to use this code as the basis of your own checker.

You can compile and run the start project by following steps (make sure that you have `java` and `javac` commands configured that can run in your command line environment)

```
java -cp jlex.jar JLex.Main c.jlex
```
→ make `c.jlex.java` file
```
java -cp javacup.jar java_cup.Main c.cup
```
→ make `sym.java`, `parser.java` files
```
javac -classpath jlex.jar;javacup.jar *.java
```
→ compile all `java` file
```
java -classpath .;jlex.jar;javacup.jar Checker test.c
```
→ check `test.c` file

Submission

Compress your project into a single file named **<StudentID_Assignment1>.zip/rar** and submit to our course website. Make sure that both source code and a runnable version of your program, named **`Checker.jar`**, are also included. Then we can test your program by just type:

```
java –jar Checker.jar <input_file>
```

^-^ GOOG LUCK! ^-^