COEN 175

Phase 3 - Week 2

TA

- Antonio Gigliotti: agigliotti@scu.edu
  - Office Hours: Thursday 11 - 1 PM

Extra Help/Tutoring

- Tau Beta Pi Tutoring
  - Wednesday 2:30 - 3:30 PM
- Link to Tutoring schedule and zoom link
  - https://sites.google.com/scu.edu/scutaubetapi/tutoring?authuser=1&pli=1

Rest of Phase 3

1. Write Symbol and Scope classes
2. Write checker functions

- Due May 2nd 11:59PM

But first … Did you break your parser?

- Run the phase 2 examples and test cases and make sure your parser can still parse!
- You should not need to modify parser.cpp for the rest of phase 3

1. Write Symbol and Scope Classes

- Symbol.h given in class
- Write Symbol.cpp
  - A symbol is simply a *name* and a *Type*
- Write Scope.h & Scope.cpp
  - If using the Linked List method:
    - Each scope object should have a **pointer to its enclosing scope** and a **vector of Symbol pointers**
  - Necessary member functions:
    - **insert** - add symbol to this scope (add symbol pointer to vector)
    - **find** - find and return symbol pointer within this scope (based on name)
    - **lookup** - find and return nearest symbol pointer within this scope and enclosing scopes (based on name)
    - **remove** - find and remove symbol with given name from scope (based on name)
    - accessor functions

2. Write checker.cpp

- Use global Scope pointer to reference the current scope
- Create global strings for the 4 types of errors
  - Use report function with those error strings (see next slide)
- Functions:
  - **openScope** - open new scope and set as the current scope
  - **closeScope** - close current scope and set current scope to the enclosing scope
  - **defineFunction** - ensure it hasn't been defined before and matches previous declaration if any
  - **declareFunction** - add symbol to global scope and if previously declared, make sure type matches
  - **declareVariable** - check for possible errors and add symbol to current scope
  - **checkIdentifier** - check if a name has been declared

How to use report()

- Define a static string for error reporting (in checker):
  - `static string redefined = "redefinition of '%s'";`
- When encountering an error, call report:
  - `report(redefined, name)`
  - Note: ‘name’ is replacing %s in the string

How to consume stderr

- Remember: stderr is being checked for phase 3 (NOT stdout)
  - Okay to have extra couts in code/output
- For testing, redirect stderr to a file:
  - `./scc < hello.c` **`2>hello.myerr`**
- Optionally, throw away stdout
  - `./scc < hello.c 2>hello.myerr` **`>/dev/null`**
- Compare your error file with the solutions
  - `diff hello.myerr hello.err`