COP 3402 - SPRING 2011

Final Project – Grading Criteria

EXECUTION POINTS (35 POINTS)

- *The submitted parser source code should compile, run and produce the correct output.*
- *We decided not to test any error recovery.*

- Test Case 1 (*proj1.txt*) – **10 points**
  - No units, no procedures, no arrays.
  - Mainly testing assignments and some expressions.
  - There are 5 assignments. Each correct assignment is worth 1 point. No partial credit.
  - 5 points for correct symbol table.

- Test Case 2 (*proj2.txt*) – **5 points**
  - Testing programming constructs.
  - If statement – 2 points
  - For loop – 3 points

- Test Case 3 (*proj3.txt*) – **10 points**
  - Test of units.
  - Units are reflected in symbol table. This means that there should be some indication on how aliases are handled and variables should have correct units assigned – 6 points.
  - There should be 4 warnings about mismatched units. They are worth 1 point each.

- Test Case 4 (*proj4.txt*) – **10 points**
  - This mainly tests procedures.
  - One symbol table per procedure with correct local parameters; worth 3 points each.
  - Correct call semantics – 4 points

IMPLEMENTATION DETAILS (35 + 15 POINTS)

- *We are looking for correct functionality (at least conceptually). The actual structure or naming of the functions is irrelevant.*
- *You can get up to 15 bonus points. See details in the 'Bonus' section below.*

- Symbol Table Implementation. The symbol table should differentiate between local and global variables, variables and constants and procedures. The value of constants and the base address and size of variables should be saved. – **10 points**
- Procedure parsing – **5 points**
- Recognition of casting and units / aliases – **5 points**

- Implementation of for loops. – **5 points**
- Optimizations – **10 points**
    - The assignment requires at least two simple optimizations. Each is worth 5 points.
    - Using the one from Dr. Hughes' reference solution counts as simple optimization.
    - A hard optimization is worth 10 points.
- Bonus **– Up to 15 points**
    - Each additional code optimization will be counted (each simple one is worth 5 points, a hard one earns you 10 points)
    - If you implemented a new language construct, this will earn you between 5 – 10 extra points (depending on the complexity of the construct)
    - Clever error recovery that goes beyond synchronizing tokens is worth 5 points.
    - If you implemented your own lexical analyzer and didn't use the one provided by Remo, you get 5 bonus points.

PROJECT SUMMARY (10 POINTS)

- This document should summarize the successes and shortcomings of your submission
- Length should be about 1 page

LOG BOOK (10 POINTS)

- There should be at least 9 dated entries in your log book – **9 points**
- Format of your log – **1 point**

STUDENT-PROVIDED TEST CASES (10 POINTS)

- Dr. Hughes asked for two to five sample programs that highlight the strength and weaknesses of your compiler
- Each test case is worth 5 points.
- In order to get full points, we expect to see some comments about the purpose of your test cases either in the input files themselves or in your project description.

NOTES:

- Should your program fail to compile or crash, we will spend up to 10 minutes trying to fix the problem in your code. If only a small fix is required 2 – 3 points are taken off, while a more extensive fix could result in a deduction of up to 15 points. If we cannot find the problem, you will lose all execution points.