RYERSON POLYTECHNICAL INSTITUTE

DEPARTMENT OF MATH, PHYSICS, AND COMPUTER SCIENCE

CPS 710

FINAL EXAM

FALL 92

STUDENT ID: _____

INSTRUCTIONS

Please write your student ID on this page. Do not write it or your name on any other pages.

Please answer directly on this exam.

This exam has 4 questions and is worth 30% of the course mark.

CALCULATORS ARE ALLOWED. NO OTHER AIDS ARE ALLOWED.

Question A	25
Question B	25
Question C	25
Question D	25

This exam has 5 pages including the front page

This exam has 5 pages including the front page

Part A - General Concepts - 25 points

A1 (5 points)

What is the primary difference between a compiler and an interpreter?

A2 (5 points)

Draw a syntax tree for the expression a * 5 + 6 / y

A3 (5 points)

What is a shift-reduce parser?

A4 (5 points)

What is the definition of a syntax error?

A5 (5 points)

What is the definition of a semantic error?

Part B - Internal Tables - 25 points

B1 (8 points)

List 4 operations that are performed on the identifier table part of a symbol table

B2 (5 points)

List 5 symbol table entry fields

B3 (6 points)

What is the definition of a "stack frame" in an interpreter?

B4 (6 points)

List 3 stack frame fields

Part C - Evaluation - 25 points - (this is an easy question)

- A syntax tree node has the following structure:
  typedef struct node {
    int type;
    struct node *v1, *v2, *v3, *v4;
    } node;

- a FOR structure is organised as follows:
  - type = FOR
  - v1 points to a set of statements that initialise some variables
  - v2 points to a condition tree that must evaluate to TRUE for the FOR loop to continue
  - v3 points to a set of statments that are executed inside the loop (if v2 evaluates to TRUE)
  - v4 points to an expression evaluated at the end of the FOR loop. The value of this expression will be the value of the whole FOR loop.

- You are writing int eval(node *tree), a function which evaluates a syntax tree. Eval returns an int which is either the value of the syntax tree, or NOVALUE when a syntax tree has no value. Condition trees evaluate to TRUE or FALSE. Eval is completely written except for the part which handles FOR structures.

- Write C code for the section of eval which handles FOR structures.

- **You do not need to do any memory management in your code. You can assume that the FOR expression you are evaluating is error-free.**

Part D - Grammars - 25 points

In this question, non-terminals are in upper-case, and terminals in lower-case

D1 (6 points)

Left-factor the following set of productions
E -> P | term
P -> term * P
P -> term / P

D2 (7 points)

Remove the **left recursion** from the following set of productions:
BE -> BE or B | BE and B
B -> t | f

D3 (6 points)

What is an **ambiguous** grammar?

D4 (6 points)

Why is the following set of productions **ambiguous**?
BE -> BE and BE
BE -> BE or BE
BE -> t | f