I agree to abide by the student conduct code at the University of Hawaii in all my work at UH.

Signature:____________________ Name (print):___________________

Practice Midterm Examination
ICS313 -- Programming Language Theory

Open book includes
1 Scott’s Programming Language Pragmatics, 3rd ed
2 a Lisp reference (e.g. CLQR or Siebel, or another Lisp book),
3 Perl operator reference (pink sheet handout only).
4 Emacs reference card (purple sheet handout only)

NO notes, with the possible exception of one note card (if voted yes by the class) – maximum size 4” by 6” text/graphics on one side only. Must be submitted with the exam. (It is OK to use only that much of a full-sized piece of paper).

No calculators, cell phones, computers, music players or other electronic devices.

Blank scratch paper as needed.

Instructions
Write or print clearly.

Show your solution method/reasoning (even on multiple choice and T/F)
You have 75 minutes to complete the exam. Plan your time accordingly.
There are a total of 100 points on the exam plus 20 points of extra credit.
Turn in your crib sheet and any additional pages with your exam.

Good luck!!

<table>
<thead>
<tr>
<th>Problem</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7EC</th>
<th>ECQs</th>
<th>ECcs</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max</td>
<td>15</td>
<td>10</td>
<td>15</td>
<td>15</td>
<td>20</td>
<td>25</td>
<td>20</td>
<td></td>
<td></td>
<td>100</td>
</tr>
<tr>
<td>Score</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1) (15 p) Multiple Choice – Choose the best answer for each question.

Write the letter on the line provided at the bottom.

I. Which of the following is NOT a type of programming control structure?
(a) recursion
(b) array
(c) function call
(d) conditional
(e) loop

II. The “top level” loop in Common Lisp does NOT contain:
(a) a printing step
(b) an evaluation step
(c) a storage step
(d) a reading step
(e) a prompt

III. Which of the following does NOT describe a built-in data type in Perl?
(a) array
(b) associative array
(c) list
(d) scalar
(e) none of the above

IV. Ada Byron Lovelace ____________.
   (a) worked on the ENIAC
   (b) invented COBOL
   (c) designed the difference engine
   (d) corresponded with von Neuman
   (e) none of the above

V. Larry Wall worked on
   (a) Cobol
   (b) Lisp
   (c) Perl
   (d) Prolog
   (e) None of the above

I _______ II _______ III _______ IV _______ V _______

2) (10 p) True/False Choose the best answer

   a) T/F: BNF was used to describe ALGOL.

   b) T/F: (Emacs) Parenthesis matching and automatic indentation are features

   c) T/F: A proper function has one or more side effects.

   d) T/F: Perl is more strongly typed than C.

   e) T/F: Scripting languages like Perl are more readable than Java.

a _______ b _______ c _______ d _______ e _______

3) (15 pt) Syntax and Semantics

   a) (5 pt) Given the grammar below, show a derivation for the following expression: \( Z = (X + Y) \times Z \)

\[
\begin{align*}
\langle \text{assign} \rangle & \rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle \\
\langle \text{id} \rangle & \rightarrow X | Y | Z \\
\langle \text{expr} \rangle & \rightarrow \langle \text{id} \rangle + \langle \text{expr} \rangle \\
& \quad | \langle \text{expr} \rangle \times \langle \text{id} \rangle \\
& \quad | ( \langle \text{expr} \rangle ) \\
& \quad | \langle \text{id} \rangle
\end{align*}
\]

b) (7 pts) Define a grammar in BNF to generate all palindromes of characters a-f, but no non-palindromes. E.g. the following are palindromes ada, affa, c, beb, adada, etc. Non-palindromes abcdedbad, cafe, etc.

c) (3 pts) Show the derivation of bdd in your grammar.
4)  (15 p) Perl Programming.

Write a perl program to do a frequency count of every character in the input. This is the first step in solving a substitution cipher. Treat upper and lower case letters as the same.

Print the characters and frequencies in columns, sorted 1) alphabetically, and then in 2) decreasing frequency.

Count spaces and newline characters (print something to indicate the character in your table if it is a “non-printing” character.

For example, on the file:

Programming languages are constantly evolving. A software designer’s task is to choose the best one for each project.

The characters and frequencies sorted alphabetically:

A     9
B     1
etc…

Characters sorted by frequency (highest to lowest):

17    (space)
12    E
10    O
etc....
5) (20 p) Program Comprehension. Given definitions for function mystery and macro myst2.

```
(setq a 6) (setq b 5) (setq c 11)  (setq a 6) (setq b 5) (setq c 11)
(defun mystery (arg)
    (defmacro myst2 (a b &optional (c 100))
        '(+ ,a ,c (* ,@b) c))
    (cond
        ((numberp arg) (* arg arg))
        ((stringp arg) arg)
        ((and (listp arg) arg)
            (print (car arg)) (mystery (cdr arg)))
        (t 'completely)))
```

Show printed output, side effects and return value. for A-C. Show.

Show the expansion and result for D & E.

<table>
<thead>
<tr>
<th>Side Effects</th>
<th>Prints</th>
<th>Returns</th>
</tr>
</thead>
<tbody>
<tr>
<td>A. (mystery (list &quot;foo&quot;))</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B. (mystery (quote (&quot;I&quot; &quot;love&quot; &quot;Lisp&quot;)))</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C. (mystery (car (list 313 (+ 5 7) 1)))</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Macro expansion</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>D. (myst2 1 (10 c b) a)</td>
<td></td>
</tr>
<tr>
<td>E. (myst2 c (0 9) a)</td>
<td></td>
</tr>
</tbody>
</table>
6) **(25 p) Scope.** Show what would be printed at the points noted in the program. Also fill in the ARIs on the stack below including parameters, variables, values and links. Set variables to 0 unless initialized otherwise. The calling sequence is `main → p2 → p1 → p4 → p3`.

a) **(5 p)** Circle all variables which are referenced by **non-local statements**.

Where can side effects occur (what variables defined in what modules)?

b) **(10 p)** Static scoping. Use **pass by value/result** for all parameters. Use the **left** column below. **Fill in the current values of a, b, and c in the SPrint rows.**

c) **(10 p)** Dynamic scoping. Use **pass by reference** for all parameters. Use the **right** column below. **Fill in the current values of a, b, and c in the DPrint rows.**

<table>
<thead>
<tr>
<th>static</th>
<th>dynamic</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```plaintext
program main();
var a=5, b=7, c: integer;
{
    % start main
    procedure p1(int a):
        {
            % start P1
            a := b + c;
            c := a - c;
            SPrint    a= ___ b = ___ c = ___
            DPrint    a= ___ b = ___ c = ___
            P4();} % end p1
    procedure p2(int c);
    var b = 11: integer;
    procedure p3();
    var b = 5: integer;
    {
        % start P3
        b := a - b;
        a := c + b;
        SPrint    a= ___ b = ___ c = ___
        DPrint    a= ___ b = ___ c = ___
    } % end P3
    procedure p4();
    var a = 10, b: integer;
    {
        % start P4
        b := a + c;
        SPrint    a= ___ b = ___ c = ___
        DPrint    a= ___ b = ___ c = ___
        p3();} % end p4
    {
        % begin body P2
        c := b + c;
        a := a + c;
        SPrint    a= ___ b = ___ c = ___
        DPrint    a= ___ b = ___ c = ___
        P1(b);} % end P2
    {
        % body of main
        c := b - a;
        SPrint    a= ___ b = ___ c = ___
        DPrint    a= ___ b = ___ c = ___
p2(b); } % end main
Bottom of the stack
```
7) (20 p) Extra Credit - Programming
Write both recursive (10 p) and iterative (10 p) versions of a function called \texttt{sumabs}. This function will return the sum of the absolute values of all numbers in its parameter list, while ignoring any non-numeric elements in the list. Comment your code. Do error checking.

A sample function call would look like this:
\begin{verbatim}
(sumnum (list 5 "night" 3 (list 9 -10) (quote day) -5.9 (* 10 0.999))) => 42.89
\end{verbatim}