Quiz 1 by ________________________________

COP 4020 20 Sept 89

Each problem is worth 10 points. Budget your time carefully, the last two problems may take more time than the first two problems.

1. A. Evaluate the following simple Lisp Expressions:

(i) (cons '(a) '(b)) (ii) (append '(a) '(b))

(iii) (car '((a b) c d)) (iv) (atom '(a))

(v) (cond (nil "A") (t "B"))

B. Draw the binary tree representation for x: (setq x '(((a b) c (d)) e)) and figure out the correct car cdr sequence to obtain b from x.

2. Label the statements below as to which principle best fits. (Choose from Abstraction, Automation, Labeling, Orthogonality, Regularity and Security.)

A. Make the computer/compiler/interpreter do the tedious chores.

B. Factor out the recurring patterns.

C. For example, indexing was used to illustrate this principle in Chapter 1.

D. Independent functions should be controlled by independent mechanisms.

E. Leads to the notion of binding a symbolic name to a location.

3. Consider the Lisp functions A and B below:

```
(def A (lambda (x)                 ;; x is a list
           (cond ((null x) nil)
                 ((null (cdr x)) t)
                 (t nil))))

(def B (lambda (x)                 ;; x is a list of lists
          (cond  ((null x) t)
                 ((A (car x)) (B (cdr x)))
                 (t nil))))
```

Evaluate:

(i) (A '(c))

(ii) (A '(c d))

(iii) (A '((c)))

(iv) (B '((a)(b)))

(v) (B '(((a b))))

4. Write a recursive Lisp boolean function all-atoms with one parameter x which is a list, the function returns true if x is nil or a list of atoms.

Quiz 2 by ______________________________

COP 4020 6 Oct 89

Each problem is worth 10 points. Budget your time carefully, the quiz is on the long side.

1 Terminology: Match the letter of the phrase (below) that best applies to the following terms:

___ associativity ___ aliasing ___ binding

___ call by reference ___ call by value ___ key word

___ overloading ___ precedence ___ reserve word

___ scope of a binding

A. A word that has special meaning in certain syntactic contexts.

B. A word that can't be used for a programmer-declared object.

C. Rule for grouping the same operator.

D. Rule for priority of operators.

E. A single data object with two or more names.

F. A single name denoting more than one thing within the same scope.

G. Associates names with locations or objects.

H. That region of the program in which it is visible.

I. Actual parameters are copied into the formal parameters.

J. Addresses of actual parameters are used for formal parameters.

2. For the C code in the middle:

A. Draw a contour diagram (show parameters and procedures names too).

```
int n = 35;
main()
{
  int x = 5;
  P(7);
}
P(int  y)
{
  int z = 2;
  Q(11);
}
Q(int  w)
{
/*show*/
}
```

B. Draw the stack showing the activation records, parameters (and use "RA" for return addresses) at the "show" line.

3. Grammar: Consider the rules For the strings below either give a deivation tree, or state that it cannot be devired. (Here S is the start symbol and A & L are non-terminals.)

S ::= A | L | (S , S)
L ::= (S) | (S L)
A ::= a | b | c | d

A. (a b)   B. (a , (b , (c , d)))   C. (a ((b , (c))))   D. (a (b , c))

4. Lisp

```
A. If (def partA (lambda (x)
            (cond ((atom x) x)
                  (t (cons (partA (car x)) (partA (cdr x)))))))

evaluate (partA '(a ((b) c) d) )
```

B. Here (<= n m) is true if and only if "n <= m"

```
If (def partB (lambda (x y)
            (cond ((null x) y)
                  ((null y) x)
                  ((<= (car x) (car y)) (cons (car x) (partB (cdr x) y))
                  (t (cons (car y) (partB x (cdr y)))))))

evaluate (partB '(1 5 7) '(4 9))
```

Quiz 3 by ______________________________
COP 4020 20 Oct 89

Each problem is worth 10 points. Budget your time carefully, the quiz (as always) is on the long side.

1 Fill in the blanks:
A. A coercion is a ____________ type conversion and a cast is a ____________ type conversion.
B. Algol used two forms of parameter passing, call by value and call by _________. In C, by the use of macros (#define's with parameters) one can get the same effect as parameters passed by ___________.
C. When are two types identical? The most restrictive definition is called ________________ equivalence, while the least restrictive definition is called ________________equivalence.
D. Scoping rules determine in which environment are non-local identifiers accessed. If it is in the environment of definition, it is called ____________ scoping and if it is in the environment of its caller, it is called ____________ scoping.
E. In C, Pascal and Algol, local variables are bound to absolute addresses at __________ time, while global variables are bound to absolute addresses at ____________ time.

2. Grammar: Consider the rules S ::= A | (T
A. For the strings below either T ::=) | ST | AT
give a derivation tree, or state A ::= a | b | c | d
that it cannot be devired. (Here S is the start symbol and A & T are non-terminals.)
(i). (a b) (ii). (c (d a) b)

B. Show the grammar is ambiguous. (Find two different derivation trees for some string in the language.)

```
Program Contour; Var a, b:integer;

      Procedure One(var n, m: integer); Var c: integer;

            Procedure Xyz(var y, z: integer);
            begin ***here*** end;

            Procedure Abc(var r, s: integer);
            begin Xyx(r, s) end;

      begin Abc(c, m) end;

      Procedure Two(var i, j:integer); Var d:integer;
      begin One(i, d) end;

begin Two(a, b) end.
```

3. Draw a contour diagram for the Pascal-like code above (include parameter and procedure names too).

4. Aliases: When execution reaches the line marked ***here*** in the Pascal-like code above each procedure has a activation record. Give all the aliases for the each of the variables (even if they are out of scope at this ***here*** line). Note if w "is an alias" for x, then x "is an alias" for w.

a: ______________________ b: ______________________

c: ______________________ d: ______________________

i: ______________________ j: ______________________

m: ______________________ n: ______________________

r: ______________________ s: ______________________

y: ______________________ z: ______________________

Quiz 4 by ______________________________
COP 4020 13 Nov 89

Each problem is worth 10 points. Budget your time carefully, the quiz (as always) is on the long side.

1 Terminology: Match the letter of the phrase (below) that best applies to the following terms:

___ mutual consent
___ derived type
___ package
___ position independent parameters
___ block
___ subtype
___ renames
___ name equivalence
___ default parameters
___ structural equivalence

A. Has the same values, attributes, and operations as some other type, but is distinct from it.
B. Has the same attributes and operations as some type, but is a subset of the domain.
C. When used in Ada it creates aliases.
D. Two objects are of the same type if they are declared together or with the same type identifer.
E. Two objects are of the same type if they are declared as coming from the same domain of values.
F. Actual parameters can be listed in any order.
G. Actual parameters can be omitted.
H. Can be implemented as a degenerate procedure.
I. When used in Ada it can create an abstract data type.
J. Both creator and accessor must explicitly grant access.

2. Fill in the blanks:
A. In Ada parameter modes are _______ parameters or _______ parameters or __________ parameters.
B. Both Pascal and Algol________ allow parameters which are functions. Ada __________ allow parameters which are functions.
C. Exceptions in Ada can be started with the _________ statement, the code which is then executed is determined by __________ scoping, and this code is called the exception ____________.
D. In Ada, a program can have several instances of a _________ (for example) stack package, perhaps a stack of intergers and a stack of floats, in which case the stack operations like push and pop will be ____________.

```
Procedure main is
declare c, d: float;
  Package A is
     Function X (c, d: in float) return float is begin **INNER** ... end;
     Function Y (p: in float := 0.0) return float is begin ... end;
  End A;
  Package B is
     Function X (c, d: in integer := 1) return integer is begin ... end;
     Function Y (p: in integer := 0) return integer is begin ... end;
  End B;
begin  ***HERE***  end main;
```

3. For the Ada like code above

A. (4pts) At the ***HERE*** which function would be called by

(i) Y (45)____________ (ii) Y (p=>4.5) ________

(iii) X (c,d) __________ (iv) Y () ____________

B. (2pts) At the **INNER** is it possible to "see" the variable c in main, if so write a statement which will assign to c in main the value of the parameter d in A.X.

C. (4pts) Write down four different ways in Ada to call function "+" (left, right: in colors) return colors; with red the actual parameter for left and green the actual parameter for right.

4. Project: Use C to write a recursive-descent recognizer for the grammer below. Assume token is the next CHARACTER in the input stream. Assume the function advance(); advances the token to the next character. Assume a main() which has already called advance() once. Assume the input stream has no white space or newline characters. Write the boolean functions have U (and respectively, have V) which return true or false depending on if the input is a string in U (respectively, in V). (Sort of like get_s and get_t but they return true or false instead of anything useful.) U is the start symbol, and V is the only other nonterminal.

U ::= VV | @
V ::= U$ | #

Quiz 5 by ______________________________
COP 4020 1 Dec 89

Each problem is worth 10 points. Budget your time carefully.
#include "standard_disclaimer_on_length_of_quiz"

1 Terminology: Match the letter of the phrase (below) that best applies to the following terms:

___ stack ___ heap ___ pure function
___ scope ___ extent ___ garbage collection
___ class ___ side effect ___ dangling reference
___ applicative programming language

A. Computations are done by applying functions, so there are no side-effects (referential transparency).
B. One whose output is solely determined by its input.
C. That region of the program in which a binding is visible.
D. The time during execution in which the storage used to hold a value is bound to its name.
E. Change to a non-local variable by a function or a procedure.
F. An access path to a data structure that has been destroyed (scope exceeds extent).
G. A pool of storage for objects which are created and destroyed in an unpredictable order.
H. A pool of storage for objects which are created and destroyed in LIFO order, like activation records.
I. When used in C++ it can create an abstract data type.
J. Automatic and costly recovery of storage from data objects which can no longer be referenced by tracing out all possible reference paths.

2. Project (Lisp): evaluate:
A. (pairlis '(w x y z) '(a 7 (c) nil))

B. (assoc 's '((t.7)(u lambda (x) (y))(w.s)(s.5)(t.s)(s.a)(good.doctor)))

C. (mapcar '(lambda (x) (times x (minus x 1))) '(2 4 6 8))

```
D. (label fn (lambda (x)
       (cond ((zero x) 0)(t (plus x (fn (minus x 1))))) 5)
```

3. Fill in the blanks:

A. In Ada _________ programming is done by use of tasks. In a task a select statement allows the task to execute any of its _________ which another task has called. When two tasks interact it is called a ____________.

B. In C++ the objects are called _________, while these are similar to a structure with a typedef but they can also have functions and procedures which are also called _______ in object oriented lingo.

C. Exceptions in Ada can be started with the _________ statement, the code which is then executed is determined by __________ scoping, and this code is called the exception ____________.

D. Information hiding or encapsulation is the practice of hiding information that the user doesn't need to know about the implementation of the abstraction. Two reasons in favor of this are:

4. A. The DoD is moving towards requiring Ada as its only language. Give 3 reasons in favor of such a move.

B. Give 3 reasons against such a move.

Mid Term by ______________________________
COP 4020 27 Oct 89

Each problem is worth 10 points. Budget your time carefully.

1 Terminology: Match the letter of the phrase (below) that best applies to the following terms:

___ dangling else
___ call by name
___ overloading
___ side effect
___ aliasing
___ call by text
___ subtype
___ name equivalence
___ nonterminal symbol
___ structural equivalence

A. The lefthand side of a BNF rule.
B. Associates locations with values.
C. Has the same attributes and operations as some type, but is a subset of the domain.
D. Two objects are of the same type if they are declared together or with the same type identifer.
E. A single data object with two or more names.
F. A single name denoting more than one thing within the same scope.
G. Two objects are of the same type if they are declared as coming from the same domain of values.
H. Syntactic ambiguity issue for conditional statements.
I. All applied occurences of the formal parameter are replaced by the text of the corresponding actual parameter, with any embedded identifiers left to be bound in the local environment.
J. All applied occurences of the formal parameter are replaced by the expression of the corresponding actual parameter, with any embedded identifiers bound as they were at the point of call.

2. Lisp 1: Draw binary tree representations of the following S-expressions:

(a . (b c)) (a (b . c)) (a (b c)) (()()) (nil . (nil . nil))

3. Fill in the blanks:

A. A ___________ is an implicit type conversion and a __________ is an explicit type conversion.

B. Two methods of implementing static scoping for Pascal were given in the text, one used static chaining and the other was called the ________ method. Neither method is needed in C since C does not allow nested _________.

C. The difference between begin-end pairs in Pascal and Algol: in Pascal begin-end pairs are used just to delimit _______________ statements, while in Algol they also delimit _______________ .

D. The association of a name to a location is called a ___________, while the association of a second name to the same location is called a __________ .

E. In C, Pascal and Algol, local variables are bound to absolute addresses at __________ time, while global variables are bound to absolute addresses at ____________ time.

4. For the C code in the middle:

A. Draw a contour diagram (show parameters and procedures names too).

```
int n = 35;
main()
{
  int x = 5;
  P(7);
}
P(int y)
{
  int z = 2;
  Q(11);
}
Q(int w)
{
/*show*/
}
```

B. Draw the stack showing the activation records, parameters at the "show" line. (Use "RA back to ___" for return addresses.)

```
Program MidTerm; Var i, j:integer; a: array[1..2] of integer;

      Procedure One; begin i := 2; end;

      Procedure Two; Var i:integer; begin One; end;

      Procedure Swap ( x, y:integer ); Var i, t: integer;
      begin t := x; x := y; y := t; end.

begin Two; a[1]:= 3; a[2]:= 1; j:= 2; Swap(j, A[j]);end.
```

5. For the Pascal-like code above:
A. (i) If Dynamic scoping is used when procedure two calls procedure one, which “i” is assigned the value two?

(ii) If Static scoping is used when procedure two calls procedure one, which “i” is assigned the value two?

B. If the call to Swap is call by reference, what are the values of j, a[1] and a[2] after Swap returns?

C. If the call to Swap is call by value, what are the values of j, a[1] and a[2] after Swap returns?

D. If the call to Swap is call by name, what are the values of j, a[1] and a[2] after Swap returns

6. Lisp 2.
If (def fun (lambda (x)

```
      (cond ((null x) nil)
            ((atom x) (list x))
            (t (append (fun (car x)) (fun (cdr x)))))))
```

A. Evaluate (fun '(a . b))

B. Evaluate (fun '(a nil b))

C&D. Evaluate (fun '(a ((b) c) d))

7. C errors: Point out and correct and explain five errors in the C code below (label them (a) ... (e))

```
typedef struct { int alpha; int omega } Type_t;

full_of_errors()
{
      int x = 3;  int *y;     Type_t t;    Type_t *pt;

      t.alpha = 7;       t.omega =5;       x = pt->alpha;
      x = t->alpha;      pt = &t;          y = pt;
      pt.omega = x;      t.omega = x;      (*pt).alpha = 3;
      y = &x;            *y = x;           *x = y;
}
```

a ____

b ____

c ____

d ____

e ____

8. Project: Use C to write a recursive-descent recognizer for the grammer below. Assume token is the next CHARACTER in the input stream. Assume the function advance(); advances the token to the next character. Assume the input stream has no white space or newline characters. Write the boolean functions have_W (and respectively, have_Y) which return true or false depending on if the input is a string in W (respectively, in Y). (Sort of like get_s and get_t but they return true or false instead of anything useful.) W is the start symbol, and Y is the only other nonterminal.

W ::= #Y#
Y ::= $W | @Y | %