

Exam 2 (April 3, 2014) Name: \_\_\_\_\_

## Problem 1 (30 points)

For each of the six major array categories below, give a code example in a programming language that implements that category (specify the language). Which of the six categories are *dynamic shape* arrays?

(a) **Static:**

(b) **Fixed Stack-Dynamic:**

(c) **Conformant:**

(d) **Stack-Dynamic:**

(e) **Fixed Heap-Dynamic:**

(f) **Heap-Dynamic:**

## Problem 2 (20 points)

- (a) What is the difference between **deep binding** and **shallow binding**?  
For what kind of languages is this relevant?

- (b) Define the concept of **data type**, as it is used in programming languages.

### Problem 3 (20 points)

As in the example below, enumerate other four possible binding times, and for each give an example:

(a) **Language design time:**

Example: bind operator symbol '+' to the addition operation.

(b)

(c)

(d)

(e)

## Problem 4 (25 points)

- (a) Define the concept of **lifetime** of a variable.
- (b) Depending on its lifetime, a variable may belong to one of four categories.
  1. Enumerate the four categories, and for each category give a code example in a programming language that implements that category (specify the language).
  2. For each category, also specify one advantage and one disadvantage.

## Problem 5 (30 points)

Consider the following Ada program:

```
procedure Main is
  A, C, Y, Z: Integer;
  procedure Sub1 is
    A, B, Y: Integer;
    begin -- of Sub1
    ...
  end; -- of Sub1
  procedure Sub2 is
    A, Z, W: Integer;
    procedure Sub3 is
      A, B, X, Z: Integer;
      begin -- of Sub3
      ...
    end; -- of Sub3
    begin -- of Sub2
    ...
  end; -- of Sub2
  procedure Sub4 is
    B, X, Y, Z: Integer;
    begin -- of Sub4
    ...
  end; -- of Sub4
begin -- of Main
...
end; -- of Main
```

- (a) List all the variables, along with the program units where they are declared, that are visible in the bodies of Sub1, Sub2, Sub3, and Sub4, assuming **static scoping** is used.

- (b) Given the following calling sequences and assuming that **dynamic scoping** is used, what variables are visible during the execution of the last subprogram activated? Include with each visible variable the name of the unit where it is declared.
- (b1) Main calls Sub1; Sub1 calls Sub2; Sub2 calls Sub3.
  - (b2) Main calls Sub2; Sub2 calls Sub3;
  - (b3) Main calls Sub4; Sub4 calls Sub2; Sub2 calls Sub1.

## Problem 6 (30 points)

Consider the following code snippets in Scheme. Assume no variable has been defined yet (the interpreter was just loaded). If evaluating the code results in errors, explain the cause. If the code evaluates without error, show the value printed by the interpreter.

(a) `(car (1 2 3))`

(b) `(map (lambda (x) (* x x)) '(1 2 3))`

(c) `(let ((x 2) (y x)) y)`

(d) `(let* ((x 2) (y x)) y)`

(e) `(cadr (cons 'a 'b))`

(f) `(cadr (cons 'a '(b)))`

## Problem 7 (30 points)

Circle the statements that are **true**:

1. Ada allows narrowing coercions.
2. Nameless functions can be defined in Python using lambda expressions.
3. C++'s type checking is less effective compared to Ada's.
4. C implements both row-pointer and contiguous arrays.
5. Ada allows the definition of stack-dynamic arrays.
6. Explicit heap-dynamic variables are nameless.
7. Union fields in C++ are type checked.
8. Python has no indexing mechanism for accessing slices in an array.
9. Functions in Scheme can be returned from other functions.
10. Access to array elements is faster than access to record elements.