Name_____

CS471, Programming Language Structure I
Spring, 2003

Final Examination

This is a take-home exam. You may work with others in the class on all or part of the exam., but you *must* indicate your helpers' names on your answers. If you do not, and there is evidence of copying of answers, you may receive zero for the whole exam. The total points for each question or part of a question follows it in parentheses, thus: *(12 pts)*

1

Consider the Pascal program below.

a. State the rule for setting the static pointer of an activation record to point to the correct static ancestor record on the stack. *(5 pts)*

b. Consider the Pascal code fragment below. Draw a scope diagram, and a calling diagram for the program. *(5 pts)*

c. Draw a stack diagram just before execution of the print statement in B takes place. The activation records should show, at least, variables, parameters and static and dynamic pointers. Explain, in detail, how the static pointer is set for every activation record on the stack. *(15 pts)*

```
procedure A;
  var x : integer;
  procedure B;
  begin
    print(x)
  end;
  procedure C(y : integer);
    var x : integer;
  begin
    x := y + 1;
    if y < 3 then
      C(y + 1)
    else
    begin
      B
    end
  end;
begin
  x := 0;
  C(1)
end;
```

2

a. Java has reference semantics for objects. Explain this statement by answering the questions put in comments in the program below. *(2.5 pts each)*

```
public class Box {
  private int value;
  public Box() { value = 0; }
  public void setValue(int v) { value = v; }
  public int getValue() { return value; }
}

public class BoxTest {
  static public void main(String [] args) {
    Box x = new Box();                // what happens here?
    x.setValue(7);                    // what happens here?
    Box y = x;                        // what happens here?
    y.setValue(11);                   // what happens here?
    System.out.println(x.getValue()); // what is printed?
    System.out.println(y.getValue()); // what is printed?
  }
}
```

b. If the following method is added to class Box:

```
public Object copy() {
  Box b = new Box();
  b.setValue(getValue());
  return b;
}
```

and the line in BoxTest in boldface (**`Box y = x;`**) is changed to:

```
Box y = (Box)x.copy();
```

what does the main method in BoxTest now print? *(10 pts)* [Note that all classes in Java implicitly extend Object if none is mentioned.]

3

Consider the following Scheme function:

```
(define (mystery f L)
   (cond ((null? L) '())
         ((f (head L)) (mystery f (tail L)))
         (else (cons (head L) (mystery f (tail L))))))
```

a. Write out a derivation (a trace) of the call:

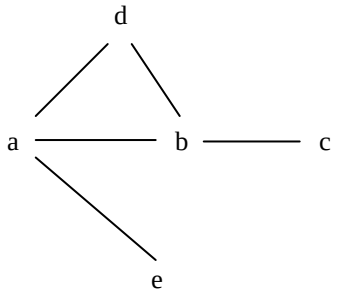
`(mystery (lambda (x) (= (modulo x 2) 1)) '(1 1 2 3 5))` *(20 pts)* [modulo is the standard arithmetic function (‘remainder’) on integers]

b. In a sentence, what does the function mystery do? *(5 pts).*

4

A graph consisting of nodes with links between them ("edges") can be described in Prolog with a set of facts about these edges. For instance the following set of facts represents the graph on the right:

```
edge(c,b).
edge(b,d).
edge(b,a).
edge(a,d).
edge(a,e).
```



A path from one node to another, following the links, may be defined by:

```
path(S,F) :- edge(S,F).
path(S,F) :- edge(F,S).
path(S,F) :- path(S,N), path(N,F).
```

a. Show, by drawing a goal tree, that Prolog can verify that there is a path from e to c. *(15 pts)* i.e. use the query:

```
?- path(e,c).
```

b. The reverse query : `path(c,e)` has a problem. What is it? *(10 pts).*

BONUS: How might the problem be fixed? *(5 pts).*