2009 November 30

Lisp Descriptive
Example Test Questions

1.

Prolog lists and Lisp lists are very similar in structure. Explain how they are similar? How they are different? How is dot-notation related to both list structures?

2.

Describe dynamic and static scoping of variables.

3.

Explain what is a lambda closure? When is a lambda closure used?

4.

Suppose we want to use the name **mapfirst** to refer to **mapcar**. Why is

```
(defmacro mapfirst (&body args) (cons 'mapcar args))
```

a better way to do this than

```
(defun mapfirst (&rest args) (apply #'mapcar args))
```

5.

B Consider the two following patterns: (a ?p ?p) and (a ?q c). Explain why you can't naively bind ?p with ?q and ?p with c to match these patterns. Propose a solution (as discussed in class) to match these two patterns.

C Explain why the solution does not work for the following two patterns:
(a ?p ?p ?p) and (a ?q ?q ?q). How does the pattern matching algorithm deal with this issue?

6.

State whether each of the following pattern pairs can be matched. If yes, give the bindings needed for the match to succeed. If no, explain why.

Pattern pairs	**Succeeds (Yes / No)**	**Bindings / Explain why not**
(a ?x (b ?z)) (a (((e))) ?y)		
(a ?x ?x) (a ?y (b ?y))		
(a ?y c)		

7.

Explain what unification means when matching patterns.

8.

Explain what is a Lisp symbol.

9.

In the context of a Lisp program what is meant by static binding.

10.

In the context of a Lisp program, what is an environment?

11.

In the context of a Lisp program what does static binding mean.

12.

What are macros? When are they used?

13.

Describe the basics of how the unification algorithm for pattern matching in Wilensky Chapter 21 works. Include a description of the data structures and various cases that can occur and how they are treated. Do not write Lisp programs.

14.

B Explain what is a property list

C Explain how we get items into a property list and how we retrieve values from a property list.

15.

What is lambda calculus and what is its relationship to Lisp?

16.

A Explain how in Lisp one can write a function that can construct a new function and then execute it. Do not describe the use of macros, this is a more general question.

B Give an example of a function A that constructs a function B, then A prints B so you can see what A constructed (nothing fancy), and finally A executes the function B.

17.

A Describe how multi-dimensional matrices would be implemented in Lisp.

18.

B How can dynamic binding be implemented in Lisp?

C How is static binding implemented in Lisp?

D Is it possible to have both static and dynamic binding in Lisp?

E If so, how can the Lisp interpreter tell them apart?

19.

Explain the difference between static and dynamic scoping.

20.

What is the funarg problem in Lisp? How does this relate to dynamic or static binding of variables?

21.

A Recursion is the method of looping in Lisp programs. Suppose you want to write a macro that requires looping, explain how you would program the loop and why you are proposing that technique.

22.

A Explain the following four results.

```
(caadaar '(('((x) (y)) ) z)          returns (x).
(caadr (caar '(('((x) (y)) ) z)))    returns (x).
(caar '(('((x) (y)) ) z))            returns '((x) (y)).
(caadr '((x) (y)))                   returns y.
```

23.

How does the database program, **retrieve**, in Wilensky Chapter 22 organize and index facts and rules. How does this reduce the time and effort in searching the database?

24.

What are an association list and a property list? What is their function in Lisp?

25.

In the context of a Lisp program, describe what is dynamic binding.

26.

In a Lisp program, describe how lambda closures are related to the environment.