Computer Science & Engineering 505 – Final

December 14, 1992

Closed Book – 110 minutes – 125 points total

Name: _____

1. (30 points) Briefly define the following words or phrases:
   (a) COMMON block
   (b) fully bracketed syntax
   (c) raise an exception
   (d) propagate an exception
   (e) metaclass
   (f) rendezvous
   (g) thunk
   (h) static link
   (i) dynamic link
   (j) `super` (in Smalltalk-80)
2. (5 points) What is the difference between single dispatching and multiple dispatching in object-oriented languages? What are the advantages and disadvantages of each technique?
3. (10 points) What is the difference between overloading, (universal) polymorphism, and generics?
4. (30 points) Suppose that the following Miranda script has been filed in.

```
my_map2 f [] [] = []
my_map2 f (a:as) (b:bs) = f a b : my_map2 f as bs
rev f x y = f y x
alligator x y = 3+y
```

   What is the result of evaluating the following Miranda expressions? (If there is a compile-time type error, or a run-time error, or a non-terminating computation, say so.) If the expression is followed by `::`, then give the type, instead of the value.

   (a) `alligator (1/0) (6/2)`
   (b) `alligator (6/2) (1/0)`
   (c) `alligator ::`
   (d) `my_map2 alligator [1..]  [10..]`
   (e) `my_map2 ::`
   (f) `my_map2 alligator ::`
   (g) `my_map2 (rev alligator) ::`
   (h) `(rev alligator) 10 20`
   (i) `rev ::`
   (j) `rev my_map2::`

5. (10 points) Miranda uses lazy evaluation for parameter passing. Suppose that programmers could optionally annotate parameters as either call-by-name or call-by-value instead.

   (a) Are there any functions for which changing the parameter to call-by-name would change the semantics of the function? If so, give an example; if not, explain why not. Are there any functions for which this change would have no effect? Again, if so, give an example; if not, explain why not. What would be the effect on performance of replacing lazy evaluation with call-by-name for a given function?

   (b) Are there any functions for which changing the parameter to call-by-value would change the semantics of the function? If so, give an example; if not, explain why not. Are there any functions for which this change would have no effect? Again, if so, give an example; if not, explain why not.

6. (10 points) Here is the standard definition for `append` in Prolog:

```
append([],Ys,Ys).
append([X|Xs],Ys,[X|Zs]) :- append(Xs,Ys,Zs).
```

   Give all the answers that would be produced for the following goals (backtracking as needed to produce more answers). If there are an infinite number of answers, give the first three or so.

   (a) `?- append(A,B,[1,2,3]).`

   (b) `?- append([1,2,3],B,C).`

   (c) `?- append(A,[7,8,9],C).`

7. (10 points) Now suppose we include a cut in the definition:

```
append_cut([],Ys,Ys) :- !.
append_cut([X|Xs],Ys,[X|Zs]) :- append_cut(Xs,Ys,Zs).
```

   Give all the answers that would be produced for similar goals:

   (a) `?- append_cut(A,B,[1,2,3]).`

   (b) `?- append_cut([1,2,3],B,C).`

   (c) `?- append_cut(A,[7,8,9],C).`

8. (5 points) What is a difference list? Why is it useful? Give several difference list representations of the list `[1,2,3]`, including the most general possible one.

9. (10 points) Compare argument passing in standard Prolog, in Flat Concurrent Prolog, and in Strand. Give examples that illustrate differences.

10. (5 points) Briefly compare the mechanism for process synchronization in Concurrent Prolog and in Strand.