Computer Science & Engineering 505 – Final

December 13, 1991

Closed Book – 110 minutes – 10 points per question – 90 points total

Name: _____

1. Describe Ada's exception handling mechanism. Be sure to include definitions of the terms "exception", "raise", "handle", and "propagate".

2. What is the difference between:

   (a) an overloaded function

   (b) a generic function

   (c) a polymorphic function

   Give an example of a language that allows overloaded functions, a language a language that allows generic functions, and a language that allows polymorphic functions.

3. Suppose `not` has been defined in Prolog in the usual way, along with an additional clause for `squid`:

```
not(X) :- call(X), !, fail.
not(X).

squid(A,A).
```

   What answers would Prolog produce for the following queries? Why?

   (a) `?- not(squid(J,K)), J=3, K=5.`

   (b) `?- J=3, not(squid(J,K)), K=5.`

   (c) `?- J=3, K=5, not(squid(J,K)).`

4. Suppose that the following Miranda script has been filed in.

```
myadd x y = x+y
mymap f [] = []
mymap f (a:x) = f a : mymap f x
crocodile = myadd 5
alligator x y = 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) `myadd ::`

   (d) `crocodile ::`

   (e) `mymap (alligator 5) ::`

5. Suppose the following COMMON LISP code has been filed in. (Recall that `defvar` declares a special variable that is looked up using dynamic scope rules; otherwise variables are looked up using lexical scoping.)

```
(setf fred1 '(a b c))
(defvar fred2 '(d e f))

(defun test1 (fred1)
  (aux1 '(x y z)))

(defun aux1 (m)
  (append fred1 m))

(defun test2 (fred2)
  (aux2 '(x y z)))

(defun aux2 (m)
  (append fred2 m))
```

What is the result of evaluating:

(a) `(test1 nil)`

(b) `(test2 nil)`

6. Suppose the following Prolog program has been filed in:

```
oyster(A,B) :- starfish(A,B).
oyster(A,B) :- starfish(B,B).
starfish(M,M).
starfish(X,3).
```

What are all the answers would Prolog produce for the query

```
?- oyster(5,Y).
```

Give them in exactly the order that Prolog would.

7. What is a difference list in Prolog? Why is it useful? Compare the use of difference lists with using the COMMON LISP `nconc` function.

8. The CLP($\mathcal{R}$) implementation attempts to satisfy constraints, or determine that they are unsatisfiable, as soon as possible. Suppose instead that all constraints were simply accumulated, and the system tried to solve them after the goal had been completely reduced.

   (a) What would be the effect on the CLP($\mathcal{R}$) semantics if this change was made? If there is no change, argue why this is the case. If there is a change, give an example of a CLP($\mathcal{R}$) program that behaves differently.

   (b) What would be the effect on efficiency? Why?

   (c) Suppose that CLP($\mathcal{R}$) used breadth-first search rather than depth-first search. Would the semantics change if the constraints were solved after the goal had been completely reduced, rather than as soon as possible? Why or why not?

9. True or false?

   (a) In Smalltalk-80, `while` loops are defined using the `whileTrue:` message, which is sent to a boolean.

   (b) In Smalltalk-80, every object is an instance of some class. Classes are themselves objects, and each class is an instance of a different metaclass. Each metaclass is an instance of a different meta-metaclass, and so forth. However, to stop the infinite regress, the system only stores 100 levels of meta-meta-meta ... metaclasses, and creates the remaining levels on demand.

   (c) The name of the Lisp `car` function stands for "contents of address register".

   (d) Variable names in Prolog have scope only within the single fact, rule, or goal in which they occur—a variable `X` in one rule is not the same as a variable `X` in another.

   (e) Parameters in Algol-60 are passed either by value or by name.

   (f) Pure Lisp doesn't include the `setq` function.

   (g) In Prolog the goal `X = 2+3` would unify `X` with the term `2+3`. In CLP($\mathcal{R}$) it would set up the constraint `X = 2+3`, which would be solved, yielding `X = 5`.

   (h) FORTRAN is not type safe, since (among other things) the `EQUIVALENCE` statement allows one to alias an integer variable and a real variable.

   (i) A FORTRAN loop `DO 100 I=1,N` will execute zero times if N=0.

   (j) Ada allows parameters to be passed using both positional and keyword notation; however, in a given call, the notations can't be mixed.