Computer Science Comprehensive Exam—Fall 2004
Programming Languages

Instructions: Do *not* put your name on the exam, please answer all the questions directly on the exam itself. You have 90 minutes. Explain answers as fully as possible; if appropriate give examples or define terms. Answer as many questions as you have time for.

1. What contributions did Dennis Ritchie, John Backus, Grace M. Hopper, *or* Alan J. Perlis make to programming?

2. Describe an object-oriented language other than C++ or Java.

3. Compute the weakest precondition for each of the following assignment statements and postconditions (please simplify):

   (a) $x := 2 * y - 4 \; \{x > 0\}$
   Answer: $2 * y - 4 > 0 \equiv 2 * y > 4 \equiv y > 2$

   (b) $b := (c + 10)/3 \; \{b > 6\}$
   Answer: $(c + 10)/3 > 6 \equiv c > 8$

   (c) $a := a + 2 * b - 1 \; \{a > 1\}$
   Answer: $a + 2 * b > 2$

   (d) $a := 2 * b + 1; b := a - 3 \; \{b < 0\}$
   Answer: $2 * b + 1 < 3 \equiv b < 1$

4. Compare and contrast the three different kinds of "constants" illustrated below.

```
const pi = 3.14159;                               (* I. Pascal.   manifest constant *)
S: constant String := Command_Line.Argument(1);-- II. Ada.
final StringBuffer sb = new StringBuffer ();  // III. Java.  single assignment
```

5. What is polymorphism? What kinds of polymorphism are there? Give an example of each kind of polymorphism in Java.

6. Write a simple program that distinguishes pass-by-reference and pass-by-value-result (copy-in/copy-out). Explain.

7. What is the difference between overloading and overriding?

8. What is the type of the following ML function? Describe in words what the function does.

```
fun pr [row] = row
 |  pr [r1::r2::rows] =
      if abs(hd r1)>=abs(hd r2) then pr(r1::rows) else pr(r2::rows)
 ;
```

9. What is the type of the following ML function? Describe in a few words what the function does.

```
fun map f nil    = nil |
    map f (h::t) = (f h) :: (map f t);
```

10. Define the member function in PROLOG that tests if an element is a member of a list.

11. Consider implementing in PROLOG a data structure of key/value pairs or a dictionary. Write a PROLOG definition for a predicate

```
Get (list, key, value)
```

where `list` is a list of pairs, `key` is an atom, and `value` is also an atom. The predicate `Get` is true when the pair of atoms `key` and `value` appears somewhere in the list. You will have to define (use) your own functor for pairs.