

Computer Science Comprehensive Exam—Fall 2011  
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. Please describe *one* of the following languages in a few sentences: D, erlang, F#, Go, or Rust.

2. Consider the following program with a loop:

```
i := 1; f := 1;
while (i <> n) {
  i := i+1; f := f*i;
}
```

(a) What does this program compute?

(b) What is the (best) loop invariant for the loop?

(c) Let  $I$  be your loop invariant in part (b). Prove the following Hoare triple is valid:

$$\{I\} \text{ i := i + 1; f := f * i } \{I\}$$

3. The programming language Java, like all object-oriented languages, has subtype polymorphism. Write a subprocedure in Java (or C#, C++) that exhibits subtype polymorphism and explain.

4. Java and C# have bounded-quantification polymorphism, an interesting new form of polymorphism which combines subtype and parametric polymorphism. Give an example and explain.

5. Describe *list comprehension* as in Python or Haskell.

6. Consider two separate, independent executions of the following Ada-like program. Assuming that X is passed by copy-in/copy-out, what are the values of I and A after the call? Assuming that X is passed by reference, what are the values of I and A after the call?

```
PP: declare
    -- declare an array of 5 elements
    A: array (1..5) of Integer := (1,2,3,4,5);
    I: Integer := 1;
    procedure P (X: Integer) is
    begin
        X := 18;  I := 2;  X := 10;
    end P;
begin
    P (A[I]); -- call P
    -- value of "I", values of "A"?
end PP;
```

7. Dynamic dispatch is key mechanism in object-oriented languages. Give a simple example, that clearly illustrates dynamic dispatch in Java or C++.

8. Consider the following Haskell function:

```
z f e nil      = e
z f e (x:xs) = f x (z f e xs)
```

(a) What is the type of the Haskell function?

(b) Describe in plain words what the function does.

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

```
datatype Tree = nl | lf of int * Tree * Tree;
fun f nl = [] | f (lf (x,l,r)) = (f l) @ (x :: (f r));
```

where `::` is the binary, infix “cons” operation and `@` binary, infix append operation on lists.

10. Find the (most general) unifying substitution for each of the following pairs of terms ( $x$ ,  $y$ , and  $z$  are variables), if it exists. To the right of each pair, write “no unifier” if none exists, otherwise give the unifying substitution.

- (a)  $g(a, c)$   $g(a, d)$
- (b)  $h(a, x, c)$   $h(a, x, d)$
- (c)  $h(c, a, x)$   $h(c, a, y)$
- (d)  $g(a, c)$   $g(c, a)$
- (e)  $g(a, b)$   $g(a, b)$
- (f)  $g(a, x)$   $g(a, h(b, c, x))$
- (g)  $g(y, x)$   $g(a, h(b, c, y))$
- (h)  $g(a, y)$   $g(a, y)$
- (i)  $g(g(a, b), h(x, a, y))$   $g(g(z, b), h(b, a, b))$
- (j)  $g(g(a, x), h(a, x, b))$   $g(g(a, b), h(a, a, b))$
- (k)  $h(z, z, z)$   $h(x, b, y)$
- (l)  $g(g(a, x), h(y, a, b))$   $g(y, x)$
- (m)  $g(g(a, x), h(y, a, b))$   $g(z, x)$

11. Formulate in PROLOG the classical syllogism:

All men are mortal;  
 Socrates is a man;  
 Therefore Socrates is mortal.