Fall 2011

PL Midterm

104 points in total

1 So far, we have already mentioned two design considerations that make Java suitable for use in the Internet. What are the two design considerations? (4%)

2 The statement "C++ is a compiled language." is problematic. What is the correct wording? (4%)

3 What are the pros and cons of dynamic scoping? (4%)

4 What are the pros and cons of dynamic type binding? (4%)

5 Consider the Javascript function below.

```
function f(n)
{
    var r="1"
    for (var i="2";i<=n;i++) { r*=i; alert(typeof i); }
    return r
}
```

a) What are the messages yielded by `alert` during the evaluation of `f(3)`? (4%)

b) What is the value of `f(10)`? of `f("10")`? (4%)

N.B. 9! = 362880

6 The following three files constitute a C++ program.

File 1.cpp

```
#include <iostream>
using namespace std;
extern int x;
int g();
static int f() { return x; }
int main()
{
    cout << f() << g();
}
```

File 2.cpp

```
int x=1;
int f()
{
    static int x=2; return x;
}
```

File 3.cpp

```
struct A { static const int x=3; }
int f();
int g() { return A::x+f(); }
```

C++ prefers replacing one of the 3 occurrences of `static` by other language construct. Which one and how? (4%)

7 The following Fortran 95 function contains two syntax errors.
Figure out the errors. (Hint: Indicate which line contains what error. Use the line numbers.)
Also, figure out the phrase (i.e. lexical, syntax, or semantic analyzer) of the compiler that detects each error. (4%)

```
function fact(n)            ! 1
integer :: n,fact           ! 2
if (n==0) then              ! 3
   fact = 1                 ! 4
else                        ! 5
   fact = n*fact(n-1)       ! 6
end                         ! 7
end                         ! 8
```

8 Show the output of the following Fortran 95 program. (4%)

```
program test
integer, dimension(2,2,2) :: a
read *, a ! 1,2
print *, a(1,:,:)
print *, a(2,:,:)
end program test
```

9 Consider the following Perl program with two blanks α and β.

```
$x=0;
sub sub2
{
   sub sub3 { __α__ $x=2; sub2(); }      # local function
   print $x;                             # free variable
   $c++; if ($c<2) { sub3; }
}
sub sub1 { __β__ $x=1; sub2; }
$c=0;
sub1;
```

Show the output of the program if (8%)

a) α = **my**, β = **my**
b) α = **my**, β = **local**
c) α = **local**, β = **my**
d) α = **local**, β = **local**

10 a) The following Perl subroutine is meant to multiply the elements of the array passed to it. Fill in the blank to complete the subroutine. (2%)

```
sub f
{
   my $r = 1;
   for (____) { $r *= $_;  }
   return $r;
}
@a=(1..5);
print f @a;     # output 120
```

b) Redo a), but this time assumes that the **for** loop is written incompletely as (2%)

```
for (my $i=0;____;$i++) { $r *= $_[$i]; }
```

11 a) The following Scheme function solves the Towers of Hanoi problem. It moves n disks from peg a to peg b with the help of the auxiliary peg c. What is the value of the call **(hanoi 2 'a 'b 'c)**? (4%)

```
(define hanoi
  (lambda (n a b c)
    (if (= n 0)
        '()
        (append (hanoi (- n 1) a c b)
                (cons `(,a . ,b)  (hanoi (- n 1) c b a))))))
```

b) Let's turn the function of part a) into an equivalent function in accumulator-passing style.

```
(define hanoiAPS
  (lambda (n a b c acc)
    (if (= n 0) acc (hanoiAPS (- n 1) a c b _____))))
```

Fill in the blank to complete the function.

Also, what should be the initial value of the parameter **acc**? (4%)

c) Let's turn the function of part a) into an equivalent function in continuation-passing style.

```
(define hanoiCPS
  (lambda (n a b c con)
    (if (= n 0) (con '()) (hanoiCPS (- n 1) a c b _____))))
```

Fill in the blank to complete the function.

Also, what should be the initial value of the parameter **con**? (4%)

12 Given

```
(define x '(a (b c) d))
(define y (cons (list x) x))
```

a) Draw a diagram showing the internal list structures bound to **x** and **y**. (4%)

b) What is the value of each expression below? (4%)

1 `y`

2 `(caddr y)`

13 a) What are the **best**-case space and time complexities of function **isort** below? (4%)

```
(define isort
  (lambda (xs)
    (if (null? xs)
        '()
        (let insert ((x (car xs))(ys (isort (cdr xs))))
          (cond ((null? ys) (list x))
                ((<= x (car ys)) (cons x ys))
                (else (cons (car ys) (insert x (cdr ys)))))))))
```

b) Redo a) for imperative-style insertion sort. (4%)

14 a) Infer the type of the recursive function fix defined by (4%)

fix = λf.f (fix f)

N.B. Using ML's notation, the function fix is defined as

```
val fix = fn f => f (fix f);
```

b) Explain why the Y combinator is untypable in ML. (4%)

N.B. Recall that Y = λf.(λx.f(xx))(λx.f(xx))

15 In this problem, we shall use the abbreviated λ-expression, say, λabcde.*exp* to denote the curried function λaλbλcλdλe.*exp*.

Now, let £ = $\lambda\underbrace{\text{abcdefghijklmnopqstuvwxyzr}}_{\text{26 English letters (Letter r at the end)}}.\text{r}(\underbrace{\text{thisisafixedpointcombinator}}_{\text{27 letters}})$

a) Prove that \$ = $\underbrace{\text{££££££££££££££££££££££££££}}_{\text{26 £'s}}$ is a fixed-point combinator with lazy evaluation. (4%)

b) Rephrase \$ as a fixed-point combinator with eager evaluation. (Proof isn't needed.) (4%)

c) Use \$ to obtain the solution to fix of the equation

fix = λf.f (fix f)

Note: It suffices to use λ-notation. No Scheme code is needed. (4%)

16 a) Compile the following λ-expression into code consisting of S, K, and I (4%)

$(\lambda x.x\ 2\ 3)\ +$

b) Reduce the compiled code of part a) with lazy evaluation (4%)

Hint

The definitions of S, K, and I as well as the compilation algorithm are given below.

$S = \lambda x.\lambda y.\lambda z.x\ z\ (y\ z)$

$K = \lambda x.\lambda y.x$

$I = \lambda x.x = S\ K\ K$

compile x	⇒	x, if x is a variable, built-in constant, or built-in function
compile (e1 e2)	⇒	compile e1 (compile e2)
compile λx.e	⇒	abstract x (compile e)
abstract x x	⇒	I
abstract x y	⇒	K y, if y is a variable (≠ x), built-in constant or built-in function
abstract x (e1 e2)	⇒	S (abstract x e1) (abstract x e2)