Tuesday 20/05/2019 Final Exam Duration: 90 minutes

Name: Student No:

P1 [20 points]

Ch8. Pr. 24

1. What is the use of a break statement in switch statements?

5 Page 362: The break stmt, which is actually a restricted goto, is normally used for exiting switch stmts. break transfers control to the first stmt after the compound stmt in which it appears.

Ch8. Pr. 11

2. What is the role of the default segment in a switch statement?

5 Page 361: The optional default segment is for unrepresented values of the control expression. If the value of the contr. expr. is not represented and no def. seg. is present, then the stmt does nothing.

3. Let's say you are given a new programming language which you didn't know before. Without looking at its compiler code, just by running some code, how would you test and understand whether it stores multidimensional arrays in row major order or column major order?

5 You can set a large 2D array and traverse it several times to see which way works faster.

Ch.8. Pr.28

4. How does a functional language implement repetition?

5 Page 372. Rather than iteration to control repetition, fp languages use recursion.

(Ch 7. Pr.19)

P2 [10 points]

Consider the C program:

```
int fun(int *i) {
    *i += 5;
    return 4;
}
void main() {
    int x = 3;
    x = x + fun(&x);
}
```

What is the value of x after the 2nd line in main, assuming

a. operands are evaluated left to right? x: 7 → 5

b. operands are evaluated right to left? x: 12 → 5

Ch 5. Prob. Set-6.

P3 [12 points]

Consider the JavaScript code:

```
// The main program
var x; //(Version A)
function sub1() {
    var x; //(Version B)
    function sub2() { . . . }
}
function sub3() { . . . }
```

Assume that the execution of this program is in the following unit order:
main calls sub1
sub1 calls sub2
sub2 calls sub3

a. Assuming static scoping, in the following, which declaration of x is the correct one for a reference to x? (Circle the correct version A or B)

i. In sub1 A (B) — 2
ii. In sub2 A (B) — 2
iii. In sub3 (A) B — 2

b. Repeat part a, but assume dynamic scoping.

i. In sub1 A (B) — 2
ii. In sub2 A (B) — 2
iii. In sub3 A (B) — 2

Ch 5. Pro. Set-12.

P4 [12 points]

Consider the program:

```
// main program
var x, y, z;
function sub1() {
var a, y, z;
. . .
}
function sub2() {
var a, b, z;
. . .
}
function sub3() {
var a, x, w;
. . .
}
```

Given the following calling sequences and assuming that dynamic scoping is used, what variables are visible during execution of the last subprogram activated? (-> means "calls") For visible variables, write the name of the function where it is declared. An answer should look like: x (main); a,b (sub2); w(sub3)

a. main->sub1->sub2 Visible: x(main) y(sub1) a,b,z(sub2)

b. main->sub1->sub3 Visible: y,z(sub1) a,x,w(sub3)

c. main->sub2->sub3 Visible: y(main) b,z(sub2) a,x,w(sub3)

d. main->sub3->sub1 Visible: x,w(sub3) a,y,z(sub1)

Ch 8. Pro. Ex. 9

P5 [10 points]

Translate the following call to Scheme's COND to C or Java and set the resulting value to y.

```
(COND
((> x 10) x)
((< x 5) (* 2 x))
((= x 7) (+ x 10))
)
```

C/Java code:

```
if (x > 10) y = x;
else if (x < 5) y = 2 * x;
else if (x == 7) y = x + 10;
```

Ch.9. Pro.5

P6 [12 points] Consider the following program written in C syntax:

```
void swap(int a, int b) {
    int temp;
    temp = a;
    a = b;
    b = temp;
}
void main() {
    int value = 2, list[5] = {1, 3, 5, 7, 9};
    swap(value, list[0]);        //position 1
    swap(list[0], list[1]);      //position 2
    swap(value, list[value]);    //position 3
}
```

For each of the following parameter-passing methods, what are all of the values of the variables value and list after each of the three calls to swap?

Variable	Pass-by-value	Pass-by-reference
value at position 1	(2)	(1)
list at position 1	1 3 5 7 9	2 3 5 7 9
value at position 2	2	1
list at position 2	1 3 5 7 9	3 2 5 7 9
value at position 3	2	2
list at position 3	1 3 5 7 9	3 1 5 7 9

P7 [24 points]

a. Show the steps in the evaluation of the following Scheme expression:

Page 673

```
(CADDAR '((A N (T ) L) Y))
```

(CAR (CDR (CDR (CAR '((A N (T) L) Y)))))

(CAR (CDR (CDR ((A N (T) L)))))

(CAR (CDR (N (T) L)))

(CAR ((T) L)) → (T)

b. What does this Scheme expression return?

Page 674.

```
(CONS '(A B) '(C D))
```

Result: ((A B) C D)

(Similar to Ch.15 Pr. 10)

c. Consider the following Scheme code.

```
(define (x lis a b)
(cond
   ((null? lis) 0)
   (else
      (cond
         ((eq? (car lis) a) (+ (x (cdr lis) a b) 1 ))
         ((eq? (car lis) b) (- (x (cdr lis) a b) 1 ))
         (else (x (cdr lis) a b))   ))))
```

i. What will be the result of the call below?

```
(x '(a t g g c g a a g t c) 'g 'c)
```

Result: 2

ii. What does this code do? In other words what does x do/find?

Function x takes a simple list lis and two atoms a, b and finds #a − #b in lis.

PBonus [15 points] Recall the definition of Fibonacci numbers:
$F(0) = 0, F(1) = 1, F(n) = F(n-1) + F(n-2)$ for $n \geq 2$
Write a Scheme function to find the nth Fibonacci number. So, when we call (fib 4) it will return 3 and (fib 7) should return 13, etc.

```
(define (fib n)
   (cond
      ((= n 0) 0)
      ((= n 1) 1)
      (else (+ (fib (- n 1)) (fib (+ n 1))))
   ))
```