Midterm 2

Due No due date **Points** 20 **Questions** 16
Available Apr 6 at 7pm - Apr 6 at 8:50pm about 2 hours **Time Limit** 62 Minutes

Instructions

- You have **60 minutes** to complete this exam.
- The exam will be stopped at **8pm**.
- This exam is closed book/closed notes; **you may not communicate with anyone** other than the instructor and/or TAs and LAs during the exam.
- Any question about the exam can be posted in campuswire; make sure it's **visible only to TAs and instructors**. We will also make broad announcements and clarifications in campuswire. You can join campuswire using **https://campuswire.com/p/GB6D343A6 (https://campuswire.com/p/GB6D343A6)** and sign-up code of 8502.
- Once you begin the exam, you must complete it within the time limit. If you logout of Canvas or close your browser after you enter the exam, the **countdown will not stop**.
- Any kind of **cheating may result in failing the course.**

This quiz was locked Apr 6 at 8:50pm.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	62 minutes	16 out of 20

Score for this quiz: **16** out of 20
Submitted Apr 6 at 8:02pm
This attempt took 62 minutes.

Question 1 — 0.5 / 0.5 pts

For the following C program, y gets 3 at the end of the main function.

```
void inc(int i) {i = i + 1;}
void main() {
  int y = 3;
  inc(y);
}
```

Correct!

- (selected) True
- False

Question 2

0.5 / 0.5 pts

(append '() '(a b c d)) produces the result of '(() a b c d).

- True
- (selected) False

Correct!

Question 3

0.5 / 0.5 pts

Higher-order functions are supported by Racket.

Correct!

- (selected) True
- False

Question 4

0.5 / 0.5 pts

In Racket, both let and let* can introduce multiple local variables.

Correct!

- (selected) True
- False

Question 5

0.5 / 0.5 pts

LISP stands for "LISt Processor".

Correct!

- (●) True
- ( ) False

Question 6

0 / 0.5 pts

When function f calls function g, function g's dynamic link points to the bottom of function f's activation record.

rrect Answer

- ( ) True

ou Answered

- (●) False

Question 7

0 / 0.5 pts

In Racket, the scope of x1 in "(letrec ((x1 e1) (x2 e2)) e)" is only e2 and e.

ou Answered

- (●) True

rrect Answer

- ( ) False

Question 8

0 / 0.5 pts

When function f calls function g, function g's static link points to the bottom of function f's activation record.

ou Answered

- (selected) True

rrect Answer

- False

Question 9

0.5 / 0.5 pts

In Ada, an out parameter can be implemented through pass by value.

- True

Correct!

- (selected) False

Question 10

0.5 / 0.5 pts

Call by value does not allow the changing of an argument's value.

Correct!

- (selected) True

- False

Question 11

3.5 / 4 pts

Answer the following questions based on the following C program:

```
int i, j, a[ 5 ]; // a is an 5 element array with indices 0-4
void swap(int x, int y) {
    int temp = x;
    x = y;
    y = temp;
}
int main () {
    for (j = 0; j < 5; j++) a[ j ] = 3;
    i = 2;
    swap(i,a[i+1]);
}
```

What are the values of i and a [3] at the end of the main function if both parameters of the
swap function are passed by

- value? i = 2 , a[3] = 3
- reference? i= 3 , a[3] = 2
- value-result? i= 3 , a[3]= 2
- name? i= 2 , a[3]= 3

Answer 1:

Correct!

2

Answer 2:

Correct!

3

Answer 3:

Correct!

3

Answer 4:

Correct!

2

Answer 5:

Correct!

3

Correct!

Answer 6:

2

ou Answered

Answer 7:

2

rrect Answer

3

Answer 8:

Correct!

3

Question 12

0 / 1 pts

Answer this question based on the following Ada program.

```
procedure Main is
  A, B : Integer;
  procedure Sub1 (C:Integer) is
    D : Integer;
  begin -- of Sub1
    ...
  end; -- of Sub1
  procedure Sub2 (E:Integer) is
    procedure Sub3 (F:Integer) is
      B, D: Integer;
    begin -- of Sub3
      Sub1(100);
    end;  -- of Sub3
  begin -- of Sub2
    Sub3(E);
  end; -- of Sub2
begin -- of Main
  Sub2(10);
end; -- of Main
```

Ada is a statically scoped language. In the above program, the Main function invokes Sub2; Sub2 invokes Sub3; and Sub3 invokes Sub1.

For this call sequence, in Sub3's activation record, what does its static link point to?

rrect Answer

u Answered

- Sub2
- Sub1
- Main
- Sub3

Question 13

1 / 1 pts

Answer this question based on the following Ada program.

```
procedure Main is
  A, B : Integer;
  procedure Sub1 (C:Integer) is
    D : Integer;
  begin -- of Sub1
    ...
  end; -- of Sub1
  procedure Sub2 (E:Integer) is
    procedure Sub3 (F:Integer) is
      B, D: Integer;
    begin -- of Sub3
      Sub1(100);
    end;  -- of Sub3
  begin -- of Sub2
    Sub3(E);
  end; -- of Sub2
begin -- of Main
  Sub2(10);
end; -- of Main
```

Ada is a statically scoped language. In the above program, the Main function invokes Sub2; Sub2 invokes Sub3; and Sub3 invokes Sub1. For this call sequence, in Sub3's activation record, what does its dynamic link point to?

Correct!

- Sub2
- Main
- Sub1
- Sub3

Question 14

2 / 2 pts

The following gives a recursive definition of a mathematical function called Mac:

Mac(n) = n-20, if n>200
= Mac(Mac(n+21)), if n<=200.

Write a Racket function called Mac that computes the above function.

Your Answer:

```
(define (Mac n) (if (> n 200) (- n 20) (Mac (Mac (+ n 21)))))
```

Question 15

2 / 3 pts

The **map** function we discussed in class maps a function across list elements:

```
(define (map f x)
    (if (null? x) '()
    (cons (f (car x)) (map f (cdr x)))))
```

Now we define a function called **maplist**

```
(define (maplist f x)
```

```
(if (null? x) '()
    (cons (f x) (maplist f (cdr x)))))
```

The difference between maplist and map is that maplist applies f to every sublist, whereas map applies f to every element. (The two function definitions differ in only the last line.)

Answer the following questions. Please clearly label your answer for each question.

(a) (1 point) What is the result of (maplist length '(2 4 6 8))?
Assume length is the list-length function.

(b) (1 point) What is the result of (maplist (lambda (xs) (map plusOne xs)) '(1 2 3 4))?
Assume that plusOne is the function that adds one to a number.

(c) (1 point) map can be defined from maplist. Define map by filling the blank below.
The correct answer is short.

```
(define (map f x)
  (maplist (lambda (x) (_______)) x))
```

Your Answer:

1.)

'(4 3 2 1)

2.)

'((2 3 4 5) (3 4 5) (4 5) (5))

3.)

map(cdr (f(x)))

(c) wrong -1

Question 16

4 / 4 pts

Write a Racket removeAll function. It takes two arguments, an element x and a list l, and it returns a list with all x-s in l removed. For example,

(removeAll 'a '(b a c a b d))

should return the list '(b c b d). Define the removeAll function by using case analysis and recursion.

Your Answer:

```
(define (removeAll x l) (if (null? l) '() (if (equal? (car l) x) (removeAll x (cdr l)) (cons (car l) (removeAll x (cdr l))))))
```

Good work!

Quiz Score: **16** out of 20