

CMSC330 Spring 2012 Final Exam

Name _____

Discussion Time	10am	11am	12pm	1pm	2pm
TA Name (circle):	Tammy	Tammy	Jane Tim	Jane Richard	Richard

Instructions

- Do not start this test until you are told to do so!
- You have 120 minutes for to take this exam.
- This exam has a total of 120 points. An average of 1 minute per point.
- This is a closed book exam. No notes or other aids are allowed.
- Answer essay questions concisely in 2-3 sentences. Longer answers are not needed.
- For partial credit, show all of your work and clearly indicate your answers.
- Write neatly. Credit cannot be given for illegible answers.

	Problem	Score	Max Score
1	Programming languages		8
2	Ruby		4
3	OCaml types & type inference		6
4	OCaml higher order & anonymous functions		6
5	Regular expressions		8
6	Context free grammars and parsing		8
7	Scoping		6
8	Parameter passing		8
9	Garbage collection		4
10	Lambda calculus & encoding		14
11	Operational semantics		10
12	Java multithreading		12
13	Ruby multithreading		22
14	Markup languages		4
	Total		120

2. (4 pts) Ruby

What is the output (if any) of the following Ruby programs? Write FAIL if code does not execute.

a. (2 pts) `a = "Captain America"` **# Output =**
 `if a =~ /[a-z]+/ then`
 `puts $1`
 `end`
 `puts $1`

b. (2 pts) `a = { }` **# Output =**
 `a["Iron"] = "Man"`
 `puts a["Iron"]`

3. (6 pts) OCaml Types and Type Inference

a. (2 pts) Give the type of the following OCaml expression

`fun x y z -> [ z ; x ; y ]` **Type =**

b. (2 pts) Write an OCaml expression with the following type

`(int -> int list) -> int` **Code =**

c. (2 pts) Give the value of the following OCaml expression. If an error exists, describe the error. The function fold is given on the next page.

`fold (fun x y -> x * y) 1 [2;3;4]` **Value =**

let rec map f l = match l with [] -> [] l (h::t) -> (f h)::(map f t)	let rec fold f a l = match l with [] -> a l (h::t) -> fold f (f a h) t
--	--

4. (6 pts) OCaml Programming

Using either map or fold and an anonymous function, write a curried function *addFirsts* which when given a list of list of ints *lst*, returns the sum of the 1st elements of all the elements of *lst*. You may assume none of the elements in *lst* are null.

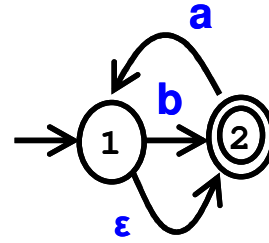
Your function must run in linear time. You may not use any library functions, with the exception of the List.rev function, which reverses a list in linear time. You may not use imperative OCaml (i.e., no ref variables).

Example:

```
addFirsts [ ] = 0
addFirsts [[1];[2];[3]] = 6
addFirsts [[1;4];[2;5;6];[3]] = 6
```

5. (8pts) Regular expressions and finite automata

Consider the following NFA.



- a. (2 pts) Give a regular expression that accepts the same set of strings as the NFA.

- b. (6 pts) Convert the NFA to a DFA using the subset construction algorithm discussed in class. Be sure to label each state in the DFA with the corresponding state(s) in the NFA.

6. (8 pts) Context free grammars and parsing

Consider the following grammar

$S \rightarrow$	AB	$ $	Ac
$A \rightarrow$	aA	$ $	epsilon
$B \rightarrow$	b	$ $	epsilon

- [illegible]

7. (6 pts) Scoping

Consider the following OCaml code.

```
let app f x = f (x+1) ;;  
let proc x = let change z = z+x in app change (x+6) ;;  
(proc 2) ;;
```

a. (2 pts) What value is returned by (proc 2) with static scoping? Explain.

b. (4 pts) What value is returned by (proc 2) with dynamic scoping? Explain.

8. (8 pts) Parameter passing

Consider the following C code.

```
int i = 2;  
void foo(int f, int g) {  
    f = f + g;  
    g = g + 2;  
}  
int main( ) {  
    int a[] = {1, 1, 1, 1};  
    foo(i, a[i-2]);  
    printf("%d %d %d %d %d\n", i, a[0], a[1], a[2], a[3]);  
}
```

a. (2 pts) Give the output if C uses call-by-value

b. (3 pts) Give the output if C uses call-by-reference

c. (3 pts) Give the output if C uses call-by-name

9. (4 pts) Garbage collection
Consider the following Java code.



```
class AvengersMovie {  
    static Avenger x, y, z;  
    private void MoviePlot( ) {  
        x = new Avenger ("Iron Man");           // object 1  
        y = new Avenger ("Black Widow");       // object 2  
        z = new Avenger ("Bruce Banner");       // object 3  
        // transformation!  
        z = new Avenger("Hulk");                // object 4  
    }  
}
```

- a. (2 pts) What object(s) are garbage when MoviePlot () returns? Explain.
- b. (2 pts) List one advantage and one disadvantage of using garbage collection.

10. (14 pts) Lambda calculus

Evaluate the following λ -expressions as much as possible.

a. (3 pts) $(\lambda x. \lambda y. y \ x \ y \ x) \ y \ x \ z$

b. (3 pts) $(\lambda x. \lambda y. \lambda z. y \ z \ x) \ (\lambda c. c) \ (\lambda a. \lambda b. b \ a) \ d$

Lambda calculus encodings

c. (8 pts) Using encodings, show $\text{succ } 2 \Rightarrow^* 3$. Show each beta-reduction.

$\Rightarrow^*$ indicates 0 or more steps of beta-reduction

$\text{succ} = \lambda z. \lambda f. \lambda y. f \ (z \ f \ y)$ $0 = \lambda f. \lambda y. y$ $1 = \lambda f. \lambda y. f \ y$ $2 = \lambda f. \lambda y. f \ (f \ y)$ $3 = \lambda f. \lambda y. f \ (f \ (f \ y))$ $4 = \lambda f. \lambda y. f \ (f \ (f \ (f \ y)))$

11. (10 pts) Operational semantics

a. (2 pts) In plain English, describe what the following means:

$(\text{fun } x = (\text{fun } y = y \ x)) \ 2 \rightarrow (x:2, \lambda y. y \ x)$

b. (8 pts) what does the expression $(\text{fun } x = (\text{fun } y = + \ x \ y)) \ 5$ evaluate to in an empty environment? In other words, find a v such that you can prove the following:

• $; (\text{fun } x = (\text{fun } y = + \ x \ y)) \ 5 \rightarrow v$

Use the operational semantics rules given in class, included here for your reference. Show the complete proof that stacks uses of these rules.

Number	$\frac{}{\bullet; n \rightarrow n}$	Lambda	$\frac{}{A; \text{fun } x = E \rightarrow (A, \lambda k. E)}$
Addition	$\frac{A; E_1 \rightarrow n \quad A; E_2 \rightarrow m}{A; + E_1 E_2 \rightarrow n + m}$	Function application	$\frac{A; E_1 \rightarrow (A', \lambda k. E) \quad A; E_2 \rightarrow v}{A, A', x:v; E \rightarrow v'}$
Identifier	$\frac{}{A; x \rightarrow A(x)}$		$A; (E_1 E_2) \rightarrow v'$

12. (12 pts) Multithreading

Consider the following multithreaded Java 1.4 code:

<pre> class Buffer { Buffer () { Object buf = null; boolean empty = true; } </pre>	<pre> void produce(o) { synchronize (buf) { 1. if (!empty) wait(); 2. empty = false; 3. notifyAll(); 4. buf = o; } } </pre>	<pre> Object consume() { synchronize (buf) { 5. if (empty) wait(); 6. empty = true; 7. notifyAll(); 8. return buf; } } </pre>	<pre> t1 = Thread.run { produce(1); } t2 = Thread.run { produce(2); } t3 = Thread.run { produce(3); } t4 = Thread.run { x = consume(); } t5 = Thread.run { y = consume(); } </pre>
---	---	--	--

Assume thread schedules are given as a list of thread name/line number/range pairs, e.g., (t1, 1), (t4, 5-8), would mean thread 1 executed line 1, followed by thread 4 executing lines 5-8.

For each of the following schedules, determine whether the schedule is possible. If it is, determine what values are assigned to x & y. If the schedule is not possible, explain why.

a. (2 pts) (t3, 1-4), (t5, 5-8), (t2, 1-4), (t4, 5-8), (t1, 1-4)

b. (3 pts) (t2, 1-4), (t3, 1), (t1, 1), (t5, 5-8), (t4, 5), (t1, 2-4), (t4, 6-8), (t3, 2-4)

c. (3 pts) (t4, 5), (t5, 5), (t1, 1-4), (t3, 1), (t2, 1), (t5, 6-8), (t4, 6-8), (t3, 2-4)

d. (3 pts) (t4, 5), (t1, 1-3), (t4, 6-8), (t1, 4), (t3, 1-4), (t5, 5-8), (t2, 1-4)



13. (22 pts) Ruby multithreading

Using Ruby monitors and condition variables, write a Ruby function `simulate(M,N)` that implements the following simulation of a superhero training facility with M heroes and N training rooms. Each hero arrives and is assigned a number between 0 and $M-1$, and each training room is assigned a number between 0 and $N-1$.

Once at the facility, each hero picks a room and trains with another superhero, repeating 10 times. Each room holds up to 2 heroes at a time. Training sessions occur with pairs of superheroes and lasts 0.01 seconds (i.e., call `sleep 0.01`). Print out a message “Room Z training X & Y” for heroes X and Y training together in each training session in room Z. The choice of a room is not significant. You may use a `getRoom()` function that returns a random room between 0 and $N-1$ to assign heroes to a room.

Each superhero must be implemented in a separate thread. You must allow pairs of heroes to train at the same time in different rooms (i.e., while one pair is calling `sleep 0.01`). Once all heroes have finished training, the simulation is complete. You may assume the simulation will automatically be terminated if all active superheroes are waiting in rooms.

You must use monitors to ensure there are no data races, and condition variables to ensure heroes efficiently wait if training rooms are all occupied. The 1st hero entering a training room will also need to efficiently wait for a 2nd hero to join the room (i.e., should wait and only wake up when the 2nd hero enters room). You may only use the following library functions.

Allowed functions:

<code>n.times { i ... }</code>	// executes code block n times, with $i = 0 \dots n-1$
<code>a = []</code>	// returns new empty array
<code>a.empty?</code>	// returns true if array a is empty
<code>a.length</code>	// returns size of array
<code>a.push(x)</code>	// pushes (adds) x to array a
<code>x = a.pop</code>	// pops (removes) element of a and assigns it to x
<code>a.each { x ... }</code>	// calls code block once for each element x in a
<code>m = Monitor.new</code>	// returns new monitor
<code>m.synchronize { ... }</code>	// only 1 thread can execute code block at a time
<code>c = m.new_cond</code>	// returns conditional variable for monitor
<code>c.wait_while { ... }</code>	// sleeps while code in condition block is true
<code>c.wait_until { ... }</code>	// sleeps until code in condition block is true
<code>c.broadcast</code>	// wakes up all threads sleeping on condition var c
<code>t = Thread.new { ... }</code>	// creates thread, executes code block in new thread
<code>t.join</code>	// waits until thread t exits

Hint: You may want to use an array `rooms[]` where `room[i] = A` for room i, where A is another array. `a.empty? = true` means room is empty. `a.push(x)` puts a hero x in the room, `a[0]` returns the 1st hero in the room, `a.pop()` takes out a hero in the room.

14. (4 pts) Markup languages

Creating your own XML tags, write an XML document that organizes the following information about the Avengers, a team of superheroes. Captain America started as a fine arts student (really!) and gained his powers from the Super Soldier Serum. The Hulk started as a nuclear physicist and gained his powers through exposure to a nuclear explosion. Iron Man started as an electrical engineer and gains his powers through his mechanical suit.

