

Multiple Choice. Write your answer to the LEFT of each problem. 3 points each

- Lisp was invented at:
  - IBM
  - MIT
  - Netscape
  - Stanford
- In C++, what operator is overloaded to provide simple input?
  - &
  - +=
  - <<
  - >>
- Which language does not allow nesting functions?
  - C
  - Scheme
  - JavaScript
  - Pascal
- (car (cdr (car (cdr '(a (b (c d e) f (g h i))))))) will result in:
  - '(g h i)
  - 'b
  - '(c d e)
  - '(g h i)
- Regular expressions are convenient for defining what element of a programming language's definition?
  - attributes
  - binary strings
  - terminals
  - tokens
- The Pascal equivalent of the C != operator is:
  - !=
  - :=
  - <>
  - ==
- Which of the following is not a JavaScript feature?
  - first-class functions
  - strong type checking
  - event-driven execution
  - integration with HTML
- What is the result of (or 'a '(b c) 'd)?
  - '(a (b c) d)
  - 'd
  - 'a
  - #t
  - #f
- Pascal indicates the value to be returned from a function by using:
  - a global variable
  - a return statement
  - a var parameter
  - the name of the function
- Which of the following allows anonymous functions?
  - C
  - Scheme
  - PL/0
  - Pascal
- If a value is second class, then it can be
  - Assigned to a variable
  - Returned from a function
  - Passed as an argument
  - All of the above
- Short-circuit evaluation has historically been missing from which language?
  - Scheme
  - JavaScript
  - Pascal
  - C
- It is considered good practice to assure that the second argument to cons is:
  - an atom
  - a function
  - a list
  - a number
- Which of the following is not a characteristic of recursive descent?
  - Error recovery
  - Small lookahead
  - Many precedence levels
  - Top-down
- What C++ language idiom avoids memory leaks?
  - Built-in strings without null terminators
  - Operator overloading
  - Resource allocation is initialization (RAII)
  - Templates

Long Answer.

- Write a Pascal function `rangeCount` to count (and return) the number of values in a global integer array `arr` (with subscripts `1..lastSub`) that are between two provided parameter *values* `low` and `high`. A value in `arr` should be counted only if it is not smaller than `low` and not larger than `high`. 18 points  

```
const lastSub=1000;
var arr: array [1..lastSub] of integer;
```
- Write a JavaScript function `rangeCount` to count (and return) the number of values in a global integer array `arr` (with subscripts `1..lastSub`) that are between two provided parameter *values* `low` and `high`. A value in `arr` should be counted only if it is not smaller than `low` and not larger than `high`. 18 points
- Write a Scheme function `rangeCount` to count (and return) the number of values in a simple integer list that are between two provided parameter values `low` and `high`. A value in the list should be counted only if it is not smaller than `low` and not larger than `high`. 18 points  
`(rangeCount '(1 4 3 2 5 4 3 6 7 3 2 1) 2 5)` will return 8

Multiple Choice:

- Write the letter of your answer on the line ( \_\_\_\_\_ ) to the LEFT of each problem.
  - CIRCLED ANSWERS DO NOT COUNT.
  - 4 points each
- For JavaScript, which expression always gives the same value as `a || b`?



2. Label the RTS locations (with PL/0 variable names) that are *accessible* at the call stop.  
(20 points)

0	var m;	0	jmp	0	58	b=55	p=17	
1		1	jmp	0	41	58	105	
1	procedure a(i);	2	jmp	0	18	57	15	ret adr
2		3	jmp	0	4	56	51	d.l.
2	const k=1000;	4	int	0	4 code for c	55	43	s.l.
2		5	lod	0	3 k	54	103	
2	procedure b(j);	6	lit	0	105	53	15	ret adr
3		7	opr	0	10 <	52	47	d.l.
3	procedure c(k);	8	jpc	0	16	51	43	s.l.
4		9	int	0	3 push args(s) for c	50	101	
4	begin	10	lod	0	3 k	49	40	ret adr
5	if k<105 then	11	lit	0	2	48	43	d.l.
8	call c(k+2)	12	opr	0	2 +	47	43	s.l.
15	else	13	int	0	-4 -(3+number of args)	46	15	
15	call stop	14	cal	1	4 c	45	29	ret adr
16	end;	15	jmp	0	17	44	39	d.l.
18		16	cal	3	-7 stop	43	23	s.l.
18	begin	17	opr	0	0 return	42	14	
19	if j<15 then	18	int	0	4 code for b	41	29	ret adr
22	call b(j+1)	19	lod	0	3 j	40	35	d.l.
29	else	20	lit	0	15	39	23	s.l.
29	begin	21	opr	0	10 <	38	13	
30	m:=i+j+k;	22	jpc	0	30	37	29	ret adr
36	call c(101)	23	int	0	3 push args(s) for b	36	31	d.l.
40	end	24	lod	0	3 j	35	23	s.l.
40	end;	25	lit	0	1	34	12	
41		26	opr	0	2 +	33	29	ret adr
41	begin	27	int	0	-4 -(3+number of args)	32	27	d.l.
42	if i<5 then	28	cal	1	18 b	31	23	s.l.
45	call a(i+1)	29	jmp	0	40	30	11	
52	else	30	lod	1	3 i	29	57	ret adr
52	call b(11)	31	lod	0	3 j	28	23	d.l.
57	end;	32	opr	0	2 +	27	23	s.l.
58		33	lit	0	1000	26	5	
58	begin	34	opr	0	2 +	25	52	ret adr
59	call a(1)	35	sto	2	5 m	24	19	d.l.
63	end.	36	int	0	3 push args(s) for c	23	1	s.l.
		37	lit	0	101	22	4	
		38	int	0	-4 -(3+number of args)	21	52	ret adr
		39	cal	0	4 c	20	15	d.l.
		40	opr	0	0 return	19	1	s.l.
		41	int	0	4 code for a	18	3	
		42	lod	0	3 i	17	52	ret adr
		43	lit	0	5	16	11	d.l.
		44	opr	0	10 <	15	1	s.l.
		45	jpc	0	53	14	2	
		46	int	0	3 push args(s) for a	13	52	ret adr
		47	lod	0	3 i	12	7	d.l.
		48	lit	0	1	11	1	s.l.
		49	opr	0	2 +	10	1	
		50	int	0	-4 -(3+number of args)	9	63	ret adr
		51	cal	1	41 a	8	1	d.l.
		52	jmp	0	57	7	1	s.l.
		53	int	0	3 push args(s) for b	6	1020	
		54	lit	0	11	5	-999999	
		55	int	0	-4 -(3+number of args)	4	-999999	
		56	cal	0	18 b	3	0	ret adr
		57	opr	0	0 return	2	0	d.l.
		58	int	0	6 code for driver	1	0	s.l.
		59	int	0	3 push args(s) for a			
		60	lit	0	1			
		61	int	0	-4 -(3+number of args)			
		62	cal	0	41 a			
		63	opr	0	0 return			

3. Give “beautiful” Scheme code for a function `levels` that will replace each atom in its single argument by its nesting level, i.e. the number of parentheses it is nested within. (20 points)

```
(levels 'a)
(levels '(a))
(levels '(((a)) b) c))
(levels '(1 (20 (3 (4 40) 3) 2) 1))
(levels '(11 (2 (3 (4 () 4) 32) 2) 15 ()))
```

would have output:

```
0
'(1)
'(((4)) 2) 1)
'(1 (2 (3 (4 4) 3) 2) 1)
'(1 (2 (3 (4 () 4) 3) 2) 1 ( ))
```

CSE 3302/5307

Name \_\_\_\_\_

Test 3

Spring 2015

Multiple Choice:

1. Write the letter of your answer on the line ( \_\_\_\_\_ ) to the LEFT of each problem.
2. CIRCLED ANSWERS DO NOT COUNT.
3. 4 points each

1. Information hiding is associated with?

- \_\_\_\_\_ A. prototypal inheritance B. data abstraction  
C. Liskov Substitution Principle D. garbage collection

2. The diamond problem is an issue with

- \_\_\_\_\_ A. continuations B. multiple inheritance  
C. interfaces D. polymorphism

3. A new property may be added to a JavaScript object using:

- \_\_\_\_\_ A. assignment B. new  
C. `Object.create()` D. prototypal inheritance

4. Continuations could be viewed as being a generalization of:

- \_\_\_\_\_ A. activation records B. combinator C. exceptions D. functions

5. In JavaScript, the result of `[0, [1, 2], "123", 3].length` will be:

- \_\_\_\_\_ A. undefined B. 3 C. 4 D. 5

6. Generational garbage collection is a generalization of:

- \_\_\_\_\_ A. mark-and-sweep B. reference counts C. Schorr-Waite D. stop-and-copy

7. Indicate the value of

```
((lambda (x y)
  (x (x y)))
 (lambda (z) (- 5 z))
 10)
```

- \_\_\_\_\_ A. 5 B. 10 C. 15 D. 20

8. What will appear on the console for the JavaScript code below?

```
arr=[];
arr[10]=1;
console.log(arr["10"]);
```

- \_\_\_\_\_ A. exception B. 5 C. 1 D. undefined

9. Type inference is associated with which language:

- \_\_\_\_\_ A. C B. JavaScript C. ML D. Pascal

10. The Y combinator is useful for:

- \_\_\_\_\_ A. anonymous functions B. continuations C. threads D. unary functions

Long Answer.

1. What is the result of executing this Scheme code? (5 points)

```
(define (f x y) (* 5 (+ x y)))
((lambda (y x z)
  (f y (x y z)))
 10
 +
 3)
```

2. What appears on the console for the code below? (10 points)

```

a={b: 5, c: 6};
b=Object.create(a);
b.c=7;
c=Object.create(b);
c.d=8;
delete c.c;
delete b.c;
console.log(c.b); _____
console.log(c.c); _____
console.log(c.d); _____
console.log(a.x); _____

```

3. What appears on the console for the code below? (15 points)

```

var makeCounterGroup = function() {
  var counter=0;
  return {
    newSub: function (initVal) {
      var subCounter=0;
      var funcs = {
        reset: function() {
          counter=counter-subCounter+initVal;
          subCounter=initVal;
        },
        up: function(val) {
          subCounter+=val;
          counter+=val;
        },
        down: function(val) {
          subCounter-=val;
          counter-=val;
        },
        value: function() { return subCounter; }
      };
      funcs.reset();
      return funcs;
    },
    value: function() { return counter; }
  };
};

```

```

var a=makeCounterGroup();
var b=makeCounterGroup();
var c=a.newSub(100);
var d=a.newSub(200);
var e=b.newSub(300);
var f=b.newSub(400);
c.up(50);
f.down(50);
e.reset();
console.log(a.value()); _____
console.log(b.value()); _____
console.log(c.value()); _____
console.log(d.value()); _____
console.log(e.value()); _____
console.log(f.value()); _____

```

4. Suppose a Pascal array is to be stored starting at location 20000 and is declared:

```
c: array[28..70,19..33,5..9] of integer;
```

If one integer takes two bytes, what is the location of c[44,22,7]? (15 points)

5. Suppose the atoms in an S-expression are integers. Write a Scheme function `prefix` to replace each atom with the sum of all atoms that have appeared to its left. (15 points)

```

> (prefix '(1 2 3 4 5))
'(0 1 3 6 10)
> (prefix '(1 (2 3) (4 ()) (((5))))))
'(0 (1 3) (6 ()) (((10))))
> (prefix '((10000) (2000 (((300)))) (40 5) (0)))
'((0) (10000 (((12000)))) (12300 12340) (12345))
> (prefix '((((1234))))))
'((((0))))

```