CSE 3302/5307 Name ____________________________________

Test 1

Summer 2013 Last 4 Digits of Mav ID # ____________________

Multiple Choice. Write your answer to the LEFT of each problem. 4 points each

1. Which of the following allows anonymous functions?
   A. C B. PL/0 C. Pascal D. JavaScript
2. Lisp was invented at:
   A. IBM B. MIT C. Netscape D. Stanford
3. Which language does not have a dangling-else ambiguity?
   A. Scheme B. JavaScript C. Pascal D. C
4. Which class of data structures is most useful for compiler symbol tables?
   A. attribute grammars B. dictionaries C. disjoint subsets D. stacks
5. Who is associated with the JavaScript language?
   A. Crockford B. Hoare C. Ritchie D. Wirth
6. Which of the following is a part of PL/0?
   A. call-by-value B. locally-restricted gotos
   C. nested recursive procedures D. simple input
7. Which of the following is true regarding attribute grammars?
   A. Inherited attributes carry information down the tree
   B. Synthesized attributes carry information down the tree
   C. They cannot represent context-sensitive information
   D. They are needed in all compilers
8. Which language's syntax would require the least effort to represent as railroad diagrams?
   A. Scheme B. JavaScript C. Pascal D. PL/0
9. PL/0 and Pascal-S are examples of which kind of semantics?
   A. attribute grammar B. denotational C. operational D. two-level grammar
10. Which language was developed most recently?
    A. C B. Scheme C. JavaScript D. Pascal

Long Answer.

1. Give Scheme code for a function `summa` to compute the summation below. *j* and *k* are positive integers. *p* is a non-negative integer. (If $j > k$, then the result is 0. Helper functions are allowed! Do NOT use math library functions) 20 points

$$\sum_{i=j}^{k} i^{p} = \texttt{(summa j k p)}$$

```
> (summa 1 5 1)
15
> (summa 1 10 2)
385
> (summa 1 1 10)
1
> (summa 3 4 2)
25
> (summa 100 200 0)
101
```

2. Give Scheme code for the `mirror` function. This function will reverse a list, but will also *recursively reverse* any nested sub-lists. If its input is an atom, then the atom is to be returned. (Helper functions are allowed!) 20 points

```
> (mirror '(a))
'(a)
> (mirror 'a)
'a
> (mirror '(a b c))
'(c b a)
> (mirror '(a b (c (d e (f g) h) i) j))
'(j (i (h (g f) e d) c) b a)
```

3. Give Pascal code for a function `summa` to compute the summation below. *j* and *k* are positive integers. *p* is a non-negative integer. All three arguments are to be passed by value. (If $j > k$, then the result is 0. Do NOT use the library functions such as `power` or `intpower`) 10 points

$$\sum_{i=j}^{k} i^{p} = \texttt{summa(j,k,p)}$$

Using your function, `writeln(summa(3,4,2))` should print `25`

4. Give JavaScript code for a function `summa` to compute the summation below. *j* and *k* are positive integers. *p* is a non-negative integer. All three arguments are to be passed by value. (If $j > k$, then the result is 0. Do NOT use library functions such as `Math.pow()`) 10 points

$$\sum_{i=j}^{k} i^{p} = \texttt{summa(j,k,p)}$$

Using your function, `alert(summa(3,4,2))` should pop up an alert box with `25`

CSE 3302 Name ______________________________________
Test 2
Summer 2013 Last 4 Digits of Mav ID # ____________________

Multiple Choice. Write your answer to the LEFT of each problem. 4 points each

1. Which of the following is not a characteristic of recursive descent?
   A. Top-down B. Error recovery C. Small lookahead D. Many precedence levels
2. Suppose the C declaration below occurs between two functions. Where would the space be allocated?
   `int arr[20000];`
   A. Static B. HeapC. Stack D. Registers
3. Which of the following would not go to the stack for the shunting-yard algorithm?
   A. (B.) C. number D. operator
4. Omitting the `new` on a call to an intended constructor will bind `this` to:
   A. the last instance created by this constructor B. the prototype
   C. an array of arguments D. the global object
5. Which of the following JavaScript objects does not have a `length`?
   A. functions B. numbers C. arrays D. strings
6. What does the brute-force recursive parsing technique store in its table?
   A. terminal symbols B. tokens C. non-terminal symbols D. grammar rules
7. Which language supports both contiguous and row-pointer methods of subscripting?
   A. JavaScript B. Pascal C. C D. Java
8. Which of the following will be treated like `false`?
   A. `5` B. `1/2` C. `NaN` D. `"          "`
9. Buddy systems are associated with which type of allocation?
   A. Static B. Stack C. HeapD. Registers
10. In C, suppose you do a `malloc()` and the provided number of bytes is larger than you requested. This is an example of:
    A. Dynamic Semantics B. External Fragmentation C. Internal Fragmentation D. Aliasing

Long Answer.

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

```
a=[11,12,13];
a["extra"]="cheese";
a.prop=[1,2,3];
a["prop"].cheese={a: a[1], b: 3.14};
a.push("me");
a["22"]=5;
console.log(a[2]);
console.log(a["prop"]["cheese"]["a"]);
console.log(a.length);
```

2. What appears on the console for the code below? (15 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);
```

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

```
var makeCounter = function(initVal,publicCounter) {
  var counter;
  var funcs= {
    reset: function() {
      counter.val=initVal;
    },
    up: function(val) {
      counter.val+=val;
    },
    down: function(val) {
      counter.val-=val;
    },
    value: function() {
      return counter.val;
    }
  };
  counter=publicCounter || {};
  funcs.reset();
  return funcs;
};

var c1=makeCounter(20,null),c2={val: 50};
c1.up(2);
c3=makeCounter(c2.val,c2);
c3.up(5);
c2.val+=100000;
c1.up(2);
var c4=makeCounter(c2.val,c2);
c2.val--;
c3.up(100);
console.log(c1.value());
console.log(c2.val);
console.log(c3.value());
console.log(c4.value());
```

4. Suppose a Pascal array is to be stored starting at location `10000` and is declared:

    ```
    c: array[10..70,30..33] of integer;
    ```

    If one `integer` takes four bytes, what is the location of `c[35,30]`? (15 points)

CSE 3302/5307 Name ____

Test 3

Summer 2013 Last 4 Digits of Mav ID # ______________________

Closed Book

Multiple Choice. Write your answer to the LEFT of each problem. 5 points each

1. 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`

2. The difference between actual parameters and formal parameters is:

    A. actuals are in the called subprogram, formals are in the caller

    B. actuals are call-by-value, formals are call-by-name

    C. actuals are in the caller, formals are in the called subprogram

    D. no difference

3. PL/0 uses static links to:

    A. Place an integer on the stack

    B. Reference data

    C. Update the display table

    D. Return from a called procedure

4. PL/0 uses dynamic links to:
    A. Place an integer on the stack
    B. Reference data
    C. Update the display table
    D. Return from a called procedure
5. (cons (cdr '(d e (g f) (a b c))) (car (cdr '(h (i j k) l m)))) will result in:
    A. '((e (g f) (a b c)) i j k)
    B. '((i j k) e (g f) (a b c))
    C. '(((g f) (a b c)) i j k)
    D. '((i j) e (g f) (a b c))
6. (cons (cdr '(e (g f) (a b c))) (car (cdr '(h (i j k) l m)))) will result in:
    A. '((e (g f) (a b c)) i j k)
    B. '((i j k) e (g f) (a b c))
    C. '(((g f) (a b c)) i j k)
    D. '((i j) e (g f) (a b c))
7. Which language's operator precedences have the least similarity to the other three?
    A. C B. Java C. JavaScript D. Pascal
8. Overloading operators may be done in which language?
    A. C++ B. Java C. JavaScript D. C
9. The value resulting from !!(4/2) in JavaScript will be
    A. true B. false C. 2 D. undefined

What is call-by-name? (5 points)

CSE 3302/5307 Name ______________________________

Test 3

Summer 2013 Last 4 Digits of Mav ID # ________________

Open Book/Notes

What is the result of executing this Scheme code? 5 points

```
(define y 4)

((lambda (x y)
   (x y))
 (lambda (y)
   ((lambda (x) (+ 8 y))
    5))
 7)
```

Long Answer. 15 points each.

1. Give Scheme code for a predicate to test whether its argument is a list consisting only of numbers in strictly ascending order (i.e. no duplicate values).

```
> (ascend 'x)
#f
> (ascend '(1 2 3 4))
#t
> (ascend '(1 (2 3) 4 5))
#f
> (ascend '(1 2 3 3 4))
#f
> (ascend '(3 2 1))
#f
> (ascend '())
#t
> (ascend '(()))
#f
```

2. Give Scheme code to replace each atom in an S-expression by the number of atoms that precede it when the S-expression is printed. The argument will always be a list - no error checking is required.

```
> (replace '(1 2 3 4))
'(0 1 2 3)
> (replace '(1 (2 3) 4 5))
'(0 (1 2) 3 4)
> (replace '((((1 (2 3) 4 5)))))
'((((0 (1 2) 3 4))))
```

```
> (replace '(3 2 1))
'(0 1 2)
> (replace '((((1 (2 3) 4 5)) ((((1 (2 3) 4 5)) (3 2 1))))))
'((((0 (1 2) 3 4)) ((((5 (6 7) 8 9)) (10 11 12)))))
> (replace '(((x (y (z) (c (b (a))))))))
'(((0 (1 (2) (3 (4 (5)))))))
```

3. Give equivalent C code (e.g. using `if` ... `else` ...) to demonstrate the short-circuit nature of C boolean operators. Do not use `&&`, `||`, or `!` in your solution! Do not use work variables!
   a. `result = a < 13 && a > 10;`
   b. `result = c < 17 || c > 20;`
   c. `result = e < 25 && !(f > 55 && g < 66);`