Midterm
CSE 131
Spring 2015

Page 1  _________ (28 points)
Page 2  _________ (24 points)
Page 3  _________ (28 points)
Page 4  _________ (28 points)
Page 5  _________ (36 points)

Subtotal  _________ (144 points = 100%)
Page 6  _________ (8 points)
Extra Credit

Total  _________

This exam is to be taken by yourself with closed books, closed notes, no electronic devices. You are allowed one side of an 8.5"x11" sheet of paper handwritten by you.
1. Given the following CUP grammar snippet (assuming all other Lexing and terminals are correct):

```plaintext
Expr ::= Expr T_PLUS {: System.out.println("A"); } Des {: System.out.println("B"); :} |
| Des {: System.out.println("C"); :} ;

Des ::= T_STAR {: System.out.println("D"); } Des {: System.out.println("E"); :} |
| T_AMPERSAND {: System.out.println("F"); } Des {: System.out.println("G"); :} |
| T_PLUSPLUS {: System.out.println("H"); } Des {: System.out.println("I"); :} |
| Des2 ;

Des2 ::= Des2 {: System.out.println("J"); } T_PLUSPLUS {: System.out.println("K"); :} |
| Des2 {: System.out.println("L"); } T_LBRACKET Expr T_RBRACKET {: System.out.println("M"); :} |
| Des2 {: System.out.println("N"); } T_DOT T_ID {: System.out.println("O"); :} |
| Des3 ;

Des3 ::= T_ID {: System.out.println("P"); :} |
| T_INT_LITERAL {: System.out.println("Q"); :} ;
```

What is the output when parsing the follow expression (you should have 21 lines/letters in your output): [21pts]

```
z[0].y++ + ++x.y[0]
```

In the above grammar, does the addition (+) operator have left-to-right or right-to-left associativity? [2pts]

If variables x and z were defined as below, what value should the expression above result in at runtime? [5pts]

```plaintext
structdef MS1 {
    int * y;
};
structdef MS2 {
    int y;
};
int a = 9;
int b;
int * c = &a;
int * d = &b;
MS1 x;
MS2 z[1];

function : void main() {
    x.y = (&c)[0];
    z[*d].y = (*c)++;
    cout << z[0].y++ + ++x.y[0];  ----> ____________________
}
2. Show the memory layout of the following C struct definition taking into consideration the **SPARC** data type memory alignment restrictions discussed in class. Fill bytes in memory with the appropriate struct member/field name. For example, if member/field name \( p \) takes 4 bytes, you will have 4 \( p \)'s in the appropriate memory locations. If the member/field is an array, use the name followed by the index number. For example, some number of \( p[0], p[1], p[2] \), etc. If the member/field is a struct, use the member name followed by its member names (e.g. \( p.a, p.b \), etc.). If the member/field is an array of structs, include both the index and member name (e.g. \( p[0].a \)). Place an X in any bytes of padding. Structs/unions are padded so the total size is evenly divisible by the strictest alignment requirement of its members. [14pts figure; 10pts questions]

```c
struct dumb {
    int         a;
    char        b[5];
};

struct by {
    short       c;
    int         d;
    char        e;
    struct dumb f;
    double *   g[2];
    char        h[4];
    double      i;
    short       j;
};

struct by lightning;
```

<table>
<thead>
<tr>
<th>low memory</th>
<th>lightning</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>high memory</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

What is the `sizeof( lightning )`? ________

What is the `offsetof( struct by, h[1] )`? ________

If `struct by` had been defined as `union by` instead, what would be `sizeof( union by )`? ________

If you rearrange the order of the struct members in `struct by` to minimize padding, what would be the size of this modified `struct by`? ________

How many bytes are saved by rearranging the `struct by` members to minimize padding? ________
3. Given the following pseudocode, determine the program output based on the specified scoping rule: [28pts]

```c
int x = 0;       -- global var declaration
int y = 0;       -- global var declaration

void sum_xy( int a, int b )
    x = x + a;
    y = y + b;

void rock()
    int y = 2; -- local var declaration
    sum_xy( 1, 1 );
    cout << x << endl;
    cout << y << endl;

void paper()
    int x = 9;  -- local var declaration
    sum_xy( 4, 8 );
    rock();
    cout << x << endl;
    cout << y << endl;

void scissors ()
    int x = 2; -- local var declaration
    int y = 9; -- local var declaration
    sum_xy( 3, 1 );
    paper();
    cout << x << endl;
    cout << y << endl;

sum_xy( 4, 5 );
rock();
paper();
scissors();
cout << x << endl;
cout << y << endl;
```

What does the program output if the language uses static scoping?

_______
_______
_______
_______
_______
_______
_______
_______
_______
_______
_______
_______
_______
_______
_______

What does the program output if the language uses dynamic scoping?

_______
_______
_______
_______
_______
_______
_______
_______
_______
_______
_______
_______
_______
_______
_______
4. Given the C array declaration on a SPARC architecture

```c
short a[2][7];
```

What is the `sizeof(a)`? ______________ [2pts]

Mark with an A all the memory location(s) where we would find the array element `a[1][6]`: [4pts]

<table>
<thead>
<tr>
<th>a:</th>
<th>lower memory</th>
<th>higher memory</th>
</tr>
</thead>
</table>

(Each box represents a byte in memory)

Given the following Reduced-C code

```c
structdef S1 { bool b; int i; float f; };
structdef S2 { bool b; int i; float f; };
structdef S3 { bool b; int i; float f; };
S1 a;
S1 b;
S2 c;
S2 d;
S3 e;
S3 f;
```

identify each of the following statements as either: [18pts]

<table>
<thead>
<tr>
<th>(A) No Error</th>
<th>(B) Compile-Time Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>____ a = b;</td>
<td>____ c = d;</td>
</tr>
<tr>
<td>____ a = c;</td>
<td>____ c = e;</td>
</tr>
<tr>
<td>____ a = d;</td>
<td>____ c = f;</td>
</tr>
<tr>
<td>____ a = e;</td>
<td>____ e = f;</td>
</tr>
<tr>
<td>____ a = f;</td>
<td>____ a.i == f.i;</td>
</tr>
<tr>
<td>____ d.f = f.i;</td>
<td>____ b.i = a.f;</td>
</tr>
</tbody>
</table>

Using the Right-Left Rule, give the C variable declaration for a variable named `sungod` that is a pointer to a pointer to a pointer to a function that takes an int and a double as two parameters and returns a pointer to an array of 4 elements where each element is a double. [4pts]
5. Given the following C code, indicate which of the 5 main areas of the C Runtime Environment each expression is located in: [18pts]

```c
int A;
static int B;

int * C ( int D, int E ) {
    static float F = 3.14;
    int G = D + E;
    return &G;
}

int main ( int argc, char *argv[] ) {
    int * H;
    H = (int *) calloc(1, sizeof(int));
    int * I = C( argc, *H );
    static int J = 5;
    int * K = &J;
    int ** L = &I;
    **L = 17;
    <------- determine runtime areas at this point in time
    free(H);
    H = NULL;
    return 0;
}
```

What programming error exists in the code above? ____________________________ [2pts]

What occurs when an object lives longer than its binding? ______________________ [2pts]

What occurs when a binding lives longer than its object? ________________________ [2pts]

Given the following C code, indicate the output for each statement, or -1 if the statement is an error: [4pts]

```c
#include <stdio.h>
void main() {
    int arr[5] = {1, 2, 3, 4, 5};
    printf("%d
", arr[3]);
    printf("%d
", arr[arr[arr[1]]]);
    printf("%d
", 4[arr];
    printf("%d
", arr[arr[1]+0[arr]]);
}
```

Identify the following C constructs as either: [8pts]

(A) Pure Declaration  (B) Definition

____ int ** foo ( int x, float y); ___ struct baz * foo( int x );
____ void foo() {} ___ float * y;
____ extern int x; ___ struct fubar ( int x; ) sl;
____ extern int a[50][50]; ___ extern int ******* p;
What gets printed when the following C program is executed? [8pts]

#include <stdio.h>
int main() {
    char a[] = "ULTRON";
    char *p = a;

    printf( "%c\n", a[5] = (*p++ = 5[a]) + 5 );
    printf( "%c\n", a[2] = (*p++ = 4[a]) - 6 );
    printf( "%c\n", p[3] = p[1] + 4 );
    printf( "%c\n", ++++p - 1 );
    printf( "%c\n", *p - 18 );
    printf( "%c\n", (*p = p[-2]) + 1 );
    printf( "%s\n", a );

    return 0;
}

A portion of the C Operator Precedence Table

<table>
<thead>
<tr>
<th>Operator</th>
<th>Associativity</th>
</tr>
</thead>
<tbody>
<tr>
<td>++ postfix increment</td>
<td>L to R</td>
</tr>
<tr>
<td>-- postfix decrement</td>
<td>L to R</td>
</tr>
<tr>
<td>[] array element</td>
<td></td>
</tr>
<tr>
<td>() function call</td>
<td></td>
</tr>
<tr>
<td>* indirection</td>
<td>R to L</td>
</tr>
<tr>
<td>++ prefix increment</td>
<td>R to L</td>
</tr>
<tr>
<td>-- prefix decrement</td>
<td>R to L</td>
</tr>
<tr>
<td>&amp; address-of</td>
<td></td>
</tr>
<tr>
<td>sizeof size of type/object</td>
<td></td>
</tr>
<tr>
<td>(type) type cast</td>
<td></td>
</tr>
<tr>
<td>* multiplication</td>
<td>L to R</td>
</tr>
<tr>
<td>/ division</td>
<td></td>
</tr>
<tr>
<td>% modulus</td>
<td></td>
</tr>
<tr>
<td>+ addition</td>
<td>L to R</td>
</tr>
<tr>
<td>- subtraction</td>
<td></td>
</tr>
<tr>
<td>.</td>
<td></td>
</tr>
<tr>
<td>= assignment</td>
<td>R to L</td>
</tr>
</tbody>
</table>