CSE 5A
Midterm
Fall 2006

Page 1 ___________ (12 points)
Page 2 ___________ (18 points)
Page 3 ___________ (34 points)
Page 4 ___________ (21 points)
Total ___________ (85 points = 81 points + 4 points EC)

This exam is to be taken by yourself with only your 1-sided notes, no electronic devices.
Operator Precedence Table

<table>
<thead>
<tr>
<th>Operators</th>
<th>Associativity</th>
</tr>
</thead>
<tbody>
<tr>
<td>- (unary) ++ -- !</td>
<td>right to left</td>
</tr>
<tr>
<td>* / %</td>
<td>left to right</td>
</tr>
<tr>
<td>+ -</td>
<td>left to right</td>
</tr>
<tr>
<td>&lt; &lt;= &gt;=</td>
<td>left to right</td>
</tr>
<tr>
<td>== !=</td>
<td>left to right</td>
</tr>
<tr>
<td>&amp;&amp;</td>
<td></td>
</tr>
<tr>
<td>= += -= *= /=</td>
<td>right to left</td>
</tr>
</tbody>
</table>

1. Using the operator precedence table above, evaluate each expression and state what gets printed.

```c
int x;
int a = 12;
int b = 7;

x = a % b * 3 - b + a;
printf( "%d\n", x );
```

(3 pts)

```c
int x;
int a = 12;
int b = 7;

x = b / 2 + a - 2 * b;
printf( "%d\n", x );
```

(3 pts)

2. What gets printed in the following blocks of statements?

```c
int a = 5;
int b = 8;
int c = 15;

if ( (b < a) || (b != 8) && (c == a+10) )
    printf( "True" );
else
    printf( "False" );
```

(3 pts)

```c
int x = 3;
int y = 2;
int z = x - 9;

if ( !(z < 6) && (x <= y) || (z < x) )
    printf( "True" );
else
    printf( "False" );
```

(3 pts)
3. Which of the following are not valid C identifiers? (Circle your answer(s).) (3 pts max) [-1 for each incorrect]

Sex-Wax cul8r double
character 24_Hour_Fitness Double

4. Fill in the blanks with the appropriate types and format specifiers to output the values correctly. (6 pts)

void main( void )
{
    __________ a = '2';
    __________ b = 15l;
    __________ c = 4.20;
    printf( "b = %____
c = %____
a = %____
", b, c, a );
}

5. What gets printed? (9 pts)

void main( void )
{
    int num = 3;
    switch ( num = num + 2 )
    {
        case 2:
            printf( "A\n" );
            num = num + 2;
            break;
        case 7:
            printf( "B\n" );
            num = num + 4;
            break;
        case 3:
            printf( "C\n" );
            num = num + num;
            break;
        case 5:
            printf( "D\n" );
            num = num + 5;
            break;
        default:
            printf( "E\n" );
            num = num + 8;
            break;
    }
    printf( "num = %d\n", num );
}
6. Write an equivalent **while loop** for the following **for loop**. (12 pts)

```c
Equivalent while loop
for (i = -12; i <= j; ++i )
{
    printf( "%d %d\n", i, j );
    j = i + j;
}
```

7. What gets printed in the following block of statements? (8 pts)

```c
#define SIZE 8
int i;
int array[SIZE] = { 2, 7, -4, 5, 16, 0, -2, 4 };;
for ( i = 0; i < SIZE; ++i )
    if ( array[i] >= 4 )
        printf( "%d\n", array[i] );
```

8. (14 pts)

```c
#include <stdio.h>
#define SIZE 8
void function1( int var1, int var2 );

int main( void )
{
    int a[SIZE] = { 2, 3, -4, 5, 16, 0, -2, 4 };;
    int z;
    scanf( "%d", &z ); /* Read an integer as input */
    if ( a[z] > a[z-1] )
    {
        z = 4;
        function1( a[z], z );
    } else {
        z = 1;
        function1( z, a[z] );
    }

    return 0;
}
void function1( int var1, int var2 )
{
    int i = 0;
    do {
        printf( "%d\n", var1 );
        ++var1;
        ++i;
    } while ( i < var2 );
}```
9. What gets printed? (16 pts)

```c
#include <stdio.h>

#define SIZE 7

int jenny( int x );

int main( void )
{
    int array[SIZE];
    int i;
    for ( i = 0; i < SIZE; ++i )
    {
        array[i] = jenny( i );
    }
    for ( i = 0; i < SIZE; ++i )
    {
        printf( "%d\n", array[i] );
    }
    printf( "i = %d\n", i );
    return 0;
}

int jenny( int x )
{
    int number[] = { 3, 0, 9, 8, 6, 7, 5 };
    if ( (x + 3) < SIZE )
        return ( number[x + 3] );
    else
        return ( number[(x + 3) - SIZE] );
}
```

Use the following to answer the questions below: (5 pts)

A) Return Type  
B) Formal Parameter  
C) Function Prototype (Function Declaration)  
D) Actual Argument  
E) Local Variable  
F) Function Definition  
G) C Preprocessor Directive

What is \( x \) in line 29?  

What is \( \text{number} \) in line 31?  

What is the \#define in line 3?  

What is the \text{int} in line 7?  

What is line 5?
Scratch Paper