Midterm Exam 1

Held on 22th July, 2011

COMPUTER PROGRAMMING FOR ENGINEERING AND SCIENCE
(CSCE 155M, SUMMER 2011)

Name :
Course No : **CSCE155M**

Instructions:

1. This is open book, open note, but not open neighbor.
2. If you have a question about the meaning of an exercise, ask! Getting things wrong because of misunderstandings can be aggravating for me as well as you.
3. If answers do not fit in the space provided, use the back of a sheet and very carefully indicate and label this so I don't miss it in grading.
4. You may take the entire period.

1. (10 points) Keeping in mind that a logical array is utilized quite differently from a numeric array, consider the follow interaction:

```
>> x = [3 5 1 7 6 2 4]
x =
     3     5     1     7     6     2     4
>> y = x > 0
y =
     1     1     1     1     1     1     1
>> z = double(y)
z =
     1     1     1     1     1     1     1
```

   (a) What is the result of entering the following three expressions?

   ```
   >> x(y)

   >> x(z)

   >> x(x)
   ```

2. (10 points) Consider the following code.

   (a) Give a formula for the final value to show how it can be calculated.

```
count = 0;
for a = 1:A
   for b = 1:B
      count = count + 1;
   end
   for b = 1:B
      count = count + 1;
   end
   for c = 1:C
      for d = 1:D
         for e = 1:E
            count = count+1;
         end
         for f = 1:F
            count = count+1;
         end
      end
   end
end
```

3. (10 points) What is output by the following fragment of code?

```
for a = 5:-2:1
   for b = a:2:5
      for c = a:b
         fprintf('%1d %1d %1d\n', a, b, c)
      end
      fprintf('tee hee!\n')
   end
   fprintf('giggle!\n')
end
```

4. (10 points) Convert each of the following to the missing forms.

   (a) (postfix:) 2 3 4 * 5 6 * + +

   (b) (prefix:) + * + 2 3 4 5

5. (10 points) Rewrite as Matlab expressions and add parentheses to show how Matlab would calculate the following expressions.

   (a) $3 + 4 * 2 + 6/3/2^2 - 5^2$

   (b) $3 > 8 + 2 || 4 == 2 + 5 * 6$

   (c) $3 > 9 | -5 <= 2 | 1 < 7 \& 2 < 5$

6. (10 points) Consider the following logic expressions.

   (a) Check the expressions in which short circuiting is enabled.
   (b) Circle the expressions in which short circuiting actually occurs.
   (c) Evaluate each expression.
   (d) Where does a side effect of a function take place, and what happens?

   Here are the expressions:

   (a) $25 > 357 || true$

   (b) $-5 == 8 \&\& 21 > 9$

   (c) $45 < 24 | logical(fprintf('howdy'))$

   (d) $45 < 245 | logical(fprintf('howdy'))$

   (e) $45 < 245 || logical(fprintf('howdy'))$

   (f) $true || 2 > 71$

   (g) $2 - 16 < 6 \& 5 > 2$

7. (10 points) Write a piece of code that determines if *grade* is 'A', 'B', etc. and prints out an appropriate message for each. Do this in each of the following ways.

   (a) Separate *if* statements
   (b) One nested *if* statement (no *elseif*'s are allowed)
   (c) An *if* with *elseif*'s
   (d) A *switch* statement

8. (10 points) Time complexity of algorithms is the subject of this exercise!

   (a) In the selection sort algorithm (which is quadratic in complexity) there are repeated rounds of searching for the largest (or smallest) value in the section that is not yet sorted. The search algorithm is similar used to find that extreme value is very similar to sequential search. Why not use something similar to the binary search that is dramatically faster?

   (b) How fast is the binary search algorithm? Explain.

   (c) Until we got into looping, we really did not focus on time complexity. Why not?

   (d) Most sorting algorithms are quadatic in complexity. What is most likely the reason?

Question	Points	Score
1	10	
2	10	
3	10	
4	10	
5	10	
6	10	
7	10	
8	10	
9	10	
10	10	
Total:	100	