

Key

## Final Exam

COMPUTER PROGRAMMING FOR ENGINEERING AND SCIENCE  
Held on 9th of August, 2012 (CSCE 155M, SUMMER 2012)

---

Name :

Save : YES NO

Course No : CSCE155M Matlab

### Instructions:

1. This is open book, open note, open computer, but not open neighbor. Please do not use email, texting, etc. during the exam.
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 you wish to have the exam saved so that you can retrieve it later, please indicate this. Exams so marked will be saved until at least the second week of classes in the fall.
4. Clearly cross off any one exercise on the exam and replace it with the following: Please describe what is most exciting about your major area of study. Why did you enter that major? How do you promote it to others?

1. (10 points) Consider the following three snippets of code. How will each respond to the input  $3 * \sin(\pi/2)$ ? Explain.

```
x = input('Enter an expression: ') evaluates
```

*x = 3*

```
y = eval(input('Enter an expression: ', 's')) eval evaluates a string
```

*y = 3*

```
z = input('Enter an expression: ', 's') 's' keeps as a string
```

*z = '3 \* sin(pi/2)'*

2. (10 points) Consider the following functions. What would be the response to the command line  $x = \text{nifty}(2+3,3) * 4$ .

*x = 8788* <sup>2197</sup> *(or 4 \* 13<sup>3</sup>)*

```
function a = nifty(b,c)
13 t = 5 b + 8 neato(5 b, 3 c);
133 a = 169 t * 13 bummer(t);
end
```

```
function x = 8 neato(5 y, 3 z)
8 x = 5 y + 3 z;
end
```

```
function x = 169 bummer(13 y)
169 x = 133 y^2;
end
```

$3+2i$	hello	hello	hello
$\begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix}$	good	bad	
$\begin{matrix} 1 & 2 \\ 4 & 4 \end{matrix}$	good	bad	
$\begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix}$			good bad

3. (10 points)

(a) Describe or sketch the structure of *cella* after the following three lines are run.

*nested*

(b) How could one access each of the *bad* messages in the cell array?

*cella{2,2}{2}    cella{1,13}{2,23}{2}    cella{1,13}{1,13}{2,2}{2}*

*(could replace {2,23} with {13})*

(c) How could one access the most deeply nested [3 4] in the cell array?

*cella{1,13}{1,13}{2,13}(2,1)*

```
cella = {3+2i, 'hello'; [1 2; 3 4], {'good', 'bad'}};
cella{1,1} = cella;
cella{1,1} = cella;
```

4. (10 points) First create an anonymous function **biggy(a,b)** that responds with **true** if the first argument is bigger than the second. Then invoke **isSorted** to determine the requests that follow:

Define **biggy** here:

*biggy = @(a,b) a > b*

*function x = biggy(a,b)  
x = a > b  
partial credit*

- (a) a random array is sorted in decreasing order

*isSorted(x, biggy)*

- (b) a random array is sorted in increasing order

*isSorted(flipr(x), biggy)*

- (c) a random array is not sorted in either decreasing or increasing order

*~isSorted(x, biggy) && ~isSorted(flipr(x), biggy)*

```
function s = isSorted(a, cFun)
    s = true;
    for ii = 2:length(a)
        if ~cFun(a(ii-1), a(ii))
            s = false;
        end
    end
end
```

5. (10 points) Consider the following function definition (next page). Now picture yourself at the command window, needing to use only that function to calculate the polynomial  $(6x^2) - 3x + 7$ .

- (a) What do you need to type to do this using just one expression?

*mOp(mOp(mOp(6, mOp(x, 2, '^'), '\*'), mOp(3, x, '\*'), '-'), 7, '+')*

- (b) Repeat, assuming the arguments to *mOp* are in the order *a, op, b*.

*mOp(mOp(mOp(6, '^', mOp(x, '^', 2)), '-'), mOp(3, '\*', x)), '+', 7)*

- (c) Repeat, assuming the arguments to *mOp* are in the order *op, a, and b*.

*mOp('+', mOp('-', mOp('\*', 6, mOp('^', x, 2)), mOp('\*', 3, x)), 7)*

- (d) Strip away all the "sugar" (that is, the parentheses, commas, quotes, and *mOp*'s) and list the characters that remain for each of the above. What do these results represent?

*6 x 2 ^ \* 3 x \* - 7 + postfix*

*6 \* x ^ 2 - 3 \* x + 7 infix*

*+ - \* 6 ^ x 2 \* 3 x 7 prefix*

```

function x = mOp(a, b, op)
switch op
    case '+'
        x = a + b;
    case '-'
        x = a - b;
    case '*'
        x = a * b;
    case '/'
        x = a / b;
    case '^'
        x = a ^ b;
end

```

6. (10 points) What do each of the following functions return? Describe any side affects.

(a) `a = fprintf('How great it is!')`

16  
msg displayed

(b) `b = input('Do you like Matlab?')`

user response  
msg displayed and pause

(c) `c = sprintf('This is the day!')`

msg  
—

(d) note the double return values! `[d, e] = sort([5 3 7 2])`

$d = [2 \ 3 \ 5 \ 7]$  values  
 $e = [4 \ 2 \ 1 \ 3]$  positions

(e) `f = uicontrol('style', 'editbox')`

handle to new editbox  
editbox

(f) `g = plot([1 2 3 4],[4 5 1 2])`

handle to plot  
plot

7. (10 points) which positions should be filled with the command `disp(d)` in the following recursive function so that the following things happen, or is it impossible?

- (a) Count goes up starting at 1: **1**
- (b) Count goes down ending at 1: **2**
- (c) Count goes up starting at 0: **0**
- (d) Count goes down to 0, then back up: **0, 2**
- (e) Count goes down to 0, repeats the 0, and then goes back up: **0, 3**
- (f) Count goes up starting at 0, then down ending at 0: **impossible**

```
function funny(d)
    % position 0
    if d > 0
        % position 1
        funny(d - 1)
        % position 2
    end
    % position 3
end
```

8. (10 points) Data that is subject to random noise, such as closing daily prices of the stock market, is often smoothed using what is called a *moving average*. This is done by taking the average of several, say 3, of the last days, including the new day's value. For example, the sequence 3, 6, 2, 7, 8, 4, 6 would yield  $(3+6+2)/3$ ,  $(6+2+7)/3$ ,  $(2+7+8)/3$ ,  $(7+8+4)/3$ ,  $(8+4+6)/3$ . Given a vector containing such values, give a snippet of code that calculates the 3 day moving averages. For full credit, make use of the "shifting" power of Matlab.

*(assume array x)*

*s = length(x)*

*a = (x(1:s-2) + x(2:s-1) + x(3:s-2)) / 3*

9. (10 points) Present a function that creates a pushbutton that when pushed changes its message randomly to one of 'Make my day!', 'Mommy, he pushed me!', and 'I dare you to try that again!'.

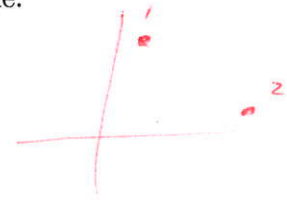
```
m = {'Make my day!', 'Mommy, he pushed me!', 'I dare you to try that again!'};
h = uicontrol('style', 'pushbutton', 'callback', '@push');
```

```
function push(s, ~)
    set(s, 'string', m{randi(3)});
end % h
```

```
⋮
```

10. (10 points) Present a function that returns the perimeter of a **rectangle**, where a **rectangle** is a struct consisting of two **points** (also structs) that contain x and y coordinates of the upper left and lower right corners of the rectangle.

```
function p = perim(r)
    dx = r.points(2).x - r.points(1).x;
    dy = r.points(1).y - r.points(2).y;
    p = 2 * (dx + dy);
end
```



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	