

Sample

Midterm Exam

COMPUTER PROGRAMMING FOR ENGINEERING AND SCIENCE
(CSCE 150E, SPRING 2010)

Held on 5th April, 2010

5 April, 2010

Name :
Course No : **CSCE150E**

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. Both sections are taking midterm exams today. You are on your honor not to reveal anything to members of the other section (assuming you are attending the 12:30 section).

1. (10 points) What is the final value of *count* in the following snippet of code? Give an algebraic formula for finding it.

```
count = 0;
for a = 1:3
    for b = 1:4
        for c = 1:2
            count = count + 1;
        end
        for d = 1:5
            count = count + 1;
        end
    end
    for e = 1:3
        count = count + 1;
    end
end
```

$$\begin{array}{r}
 3 \times 4 \times 2 = 24 \\
 3 \times 4 \times 5 = 60 \\
 + 3 \times 3 = 9 \\
 \hline
 93
 \end{array}$$

2. (10 points) What is output by the following fragment of code? What common operation is simulated by the code, assuming $b(i)$ is guaranteed to be zero? Explain briefly how the code works.

```
a = [0 0 0 5 2 3 8 6 4 7 1 7 7 5];
b = [0 0 0 3 8 5 3 9 0 9 9 8 7 3];
c = 0;
for ii = length(a): -1: 2
    s(ii) = mod(a(ii) - b(ii) + 10, 10);
    a(ii-1) = a(ii-1) - (a(ii) - b(ii) < 0);
end
fprintf('%d ', s);
```

subtraction

← (positive) difference from 0 to 9

← deduct any borrow from column to the left

1
0 0 0 ~~3~~ 8 4 7 3 7 1 9 0 2

3. (10 points) What does the following code print when it is invoked from the command line with *funone*. Be reasonably careful with the spacing. Note that *blanks* is a built-in function which generates a string with the specified number of spaces.

```
function funone()
    n = 0;
    indent(n, 'hello');
    funtwo(n+3);
    indent(n, 'ciao');
    funthree(n+3);
    indent(n, 'done');
end
```

```
function funtwo(n)
    indent(n, 'intwo');
    funthree(n+3);
    indent(n, 'outtwo');
end
```

```
function funthree(n)
    indent(n, 'inthree')
end
```

```
function indent(n, msg)
    fprintf('%s%d %s\n', blanks(n), n, msg);
end
```

0 hello
 3 in two
 6 in three
 3 out two
 0 ciao
 3 in three
 0 done

0 hello
 3 in two
 6 in three
 9 in two
 12 in three
 12 out three
 9 out two
 6 out three
 3 out two
 0 ciao
 3 in three
 6 in two
 9 in three
 12 in two
 15 in three
 15 out three
 12 out two
 9 out three
 6 out two
 3 out three
 0 done

4. (10 points) Replace *funthree* of the previous exercise with the following version. Now what is printed?

```
function funthree(n)
    indent(n, 'inthree');
    if n <= 9
        funtwo(n+3);
    end
    indent(n, 'outthree');
end
```

5. (10 points) Consider the following function definition. Now picture yourself at the command window, needing to use only that function to calculate the volume of a sphere with radius r . What do you need to type if you are permitted to enter only one line?

```
function x = multiOp(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;
    otherwise
        x = 0;
end
```

multiOp(multiOp(4, 3, '/'), multiOp(pi, multiOp(r, 3, '^'), ''), '*')*

6. (10 points) Consider the following code. Remember the trick of putting two apostrophies together to represent one apostrophe in a string? The same trick applies to percent signs in print format strings. If a is entered as 3 and b is entered as 5 and v is entered as pi, what is assigned to f ? What is printed? Be careful with spacing!

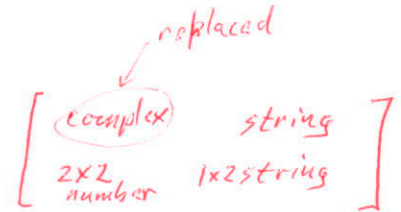
```
v = input('value: ');
a = input('left of decimal: ');
b = input('right of decimal: ');
f = sprintf('%%d.%%df', a+b+1, b); ← '%9.5f'
fprintf(f,v)
```

→ 3.14159

7. (10 points)

- (a) Describe the structure of *cella* after the first line (below) is entered.
- (b) Describe the structure of *cella* after the second line is entered. (You may simply rewrite the first line to incorporate the changes.)
- (c) In the first case, how could one access the *hello* message?
- (d) In the first case, how could one access the 4 of the array?

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



- b) $cella = \{ \{ 3+2i, 'hello', [1 2; 3 4], \{ 'good', 'bad' \} \}, 'hello'; [1 2; 3 4], \{ 'good', 'bad' \} \}$
- c) $cella \{ 1, 2 \}$
- d) $cella \{ 2, 1 \} (2, 2)$

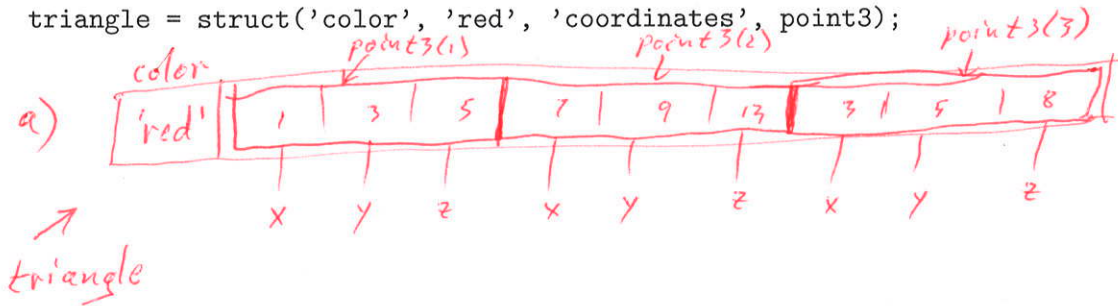
8. (10 points) Cross off any redundant (unneeded) portions of the following code. (The resulting code should always yield the same results.)

```
grade = input('Enter the grade from 0 to 100: ');
if grade < 0 || grade > 100
    disp('Invalid input')
elseif grade > 90 && grade <= 100 2
    disp('Nice A')
elseif grade > 80 && grade <= 90 2
    disp('Not a bad B')
elseif grade <= 60 && grade >= 0 2
    disp('Sorry - you blew it')
elseif grade > 60 && grade <= 70 2
    disp('Discouraging D')
elseif grade > 70 && grade <= 80 2
    disp('Average C')
end
```

9. (10 points)

- (a) Describe the structure of *triangle*. 4
- (b) Change the ~~em~~ y coordinate of *point3(2)* to 27. (Do this directly without going through *triangle*.) 3
- (c) Print the coordinates of the first point of *triangle*. 3

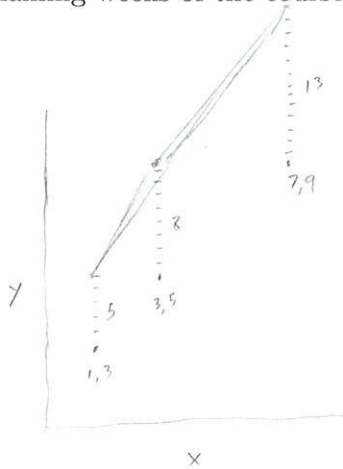
```
point3(3) = struct('x', 1, 'y', 3, 'z', 5);
point3(2) = struct('x', 7, 'y', 9, 'z', 13);
point3(1) = struct('x', 3, 'y', 5, 'z', 8);
triangle = struct('color', 'red', 'coordinates', point3);
```



b) $point3(2).y = 27$

c) $fprintf('%d', triangle.coordinates(1).x)$
 or $'' .x$
 $'' .y$
 $'' .z$

10. (10 points) Specify what parts of Matlab you would like to see emphasized in the remaining weeks of the course.



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	