L08: Special Topics

And Practice Questions

Lucas A. J. Bastien

E7 Spring 2017, University of California at Berkeley

February 03, 2017

Version: release

Announcements

Lab 03 is due on February 10 at 12 pm

Starting next week, I will drop by in lab sections to get a chance to meet you individually

Today:

- Special topics
- Many practice questions. **Review these questions on your own time and ask your GSIs if you don't understand some of them**

Next week:

- Monday: `for` and `while` loops
- Wednesday: recursion
- Friday: discussion

Practice question: nested if-statements

```
function [result] = my_nested_ifs(x, y)
if x > 3
    if y > 0
        result = 7;
        y = 2;
    else
        result = 1;
        y = 4;
    end
else
    if y == 4
        result = 3;
    else
        result = 0;
    end
end
end
```

What would the value of variable "a" be after executing the following command?

```
>> a = my_nested_ifs(5, -2);
```

(A) 7

(B) 1

(C) 3

(D) 0

(E) Matlab throws an error message

Practice question: nested if-statements

```
function [result] = my_nested_ifs(x, y)
if x > 3
    if y > 0
        result = 7;
        y = 2;
    else
        result = 1;
        y = 4;
    end
else
    if y == 4
        result = 3;
    else
        result = 0;
    end
end
end
```

What would the value of variable "a" be after executing the following command?

```
>> a = my_nested_ifs(5, -2);
```

(A) 7

(B) 1

(C) 3

(D) 0

(E) Matlab throws an error message

Practice question: more if-statements

What would the value of the variable "`delicious`" be after executing the following command?

```
>> delicious = my_yummy(1);
```

(A) 4

(B) 2

(C) 1

(D) 0

(E) -1

(F) NaN

```
1   function [yummy] = my_yummy(yummy)
2
3 -   if yummy > 0
4 -         yummy = yummy - 2;
5 -   end
6 -   if yummy < 0
7 -         yummy  = yummy + 3;
8 -   elseif yummy == 2
9 -         yummy = NaN;
10 -  else
11 -        yummy = 0;
12 -  end
```

Practice question: more if-statements

What would the value of the variable "`delicious`" be after executing the following command?

```
>> delicious = my_yummy(1);
```

(A) 4

(B) 2

(C) 1

(D) 0

(E) -1

(F) NaN

```
1   function [yummy] = my_yummy(yummy)
2
3   if yummy > 0
4         yummy = yummy - 2;
5   end
6   if yummy < 0
7         yummy  = yummy + 3;
8   elseif yummy == 2
9         yummy = NaN;
10  else
11        yummy = 0;
12  end
```

The return command

The command `return` makes the function return *i.e.* the function stops executing as if it had reached the function's `end` keyword.

What would the value of variable "a" be after executing the following command?

```
>> a = my_early_return()
```

```
1   function [result] = my_early_return()
2
3 -     result = 4;
4
5 -     return
6
7 -     result = 5;
8
9 -   end
```

(A) 4

(B) 5

The return command

The command `return` makes the function return *i.e.* the function stops executing as if it had reached the function's `end` keyword.

What would the value of variable "a" be after executing the following command?

```
>> a = my_early_return()
```

```
1   function [result] = my_early_return()
2
3 -     result = 4;
4
5 -     return
6
7 -     result = 5;
8
9 -     end
```

(A) 4

(B) 5

Checking the equality of two variables of class double

What would the value of variable "a_equals_b" be after executing the following commands?

```
>> a = 0.3;
>> b = 0.1  + 0.2;
>> a_equals_b = (a == b);
```

(A) true

(B) false

Checking the equality of two variables of class double

What would the value of variable "a_equals_b" be after executing the following commands?

```
>> a = 0.3;
>> b = 0.1  + 0.2;
>> a_equals_b = (a == b);
```

(A) true

(B) false

WHAT?? See next slide for explanation!

Checking the equality of two variables of class double

Matlab can only represent a finite number of real numbers! For example, Matlab cannot represent exactly the value 0.3:

```
>> fprintf('"0.3" with few digits: %.4f\n', 0.3)
    "0.3" with few digits: 0.3000

>> fprintf('"0.3" with many digits: %.25f\n', 0.3)
    "0.3" with many digits: 0.2999999999999999888977698

> fprintf('"0.1+0.2" with many digits: %.25f\n', 0.1+0.2)
    "0.1+0.2" with many digits: 0.3000000000000000444089210
```

Checking the equality of two variables of class double

Matlab can only represent a finite number of real numbers! For example, Matlab cannot represent exactly the value 0.3:

```
>> fprintf('"0.3" with few digits: %.4f\n', 0.3)
    "0.3" with few digits: 0.3000

>> fprintf('"0.3" with many digits: %.25f\n', 0.3)
    "0.3" with many digits: 0.2999999999999999888977698

> fprintf('"0.1+0.2" with many digits: %.25f\n', 0.1+0.2)
    "0.1+0.2" with many digits: 0.3000000000000000444089210
```

Instead of using the exact value 0.3, Matlab uses the value closest to 0.3 that it can represent

Often (especially in engineering and science applications), it is preferable to check whether two numbers are "close enough" rather than "exactly equal". It is usually not necessary, though, when, dealing with integers only

Checking the equality of two variables of class double

Matlab can only represent a finite number of real numbers! For example, Matlab cannot represent exactly the value 0.3:

```
>> fprintf('"0.3" with few digits: %.4f\n', 0.3)
    "0.3" with few digits: 0.3000

>> fprintf('"0.3" with many digits: %.25f\n', 0.3)
    "0.3" with many digits: 0.2999999999999999888977698

> fprintf('"0.1+0.2" with many digits: %.25f\n', 0.1+0.2)
    "0.1+0.2" with many digits: 0.3000000000000000444089210
```

Instead of using the exact value 0.3, Matlab uses the value closest to 0.3 that it can represent

Often (especially in engineering and science applications), it is preferable to check whether two numbers are "close enough" rather than "exactly equal". It is usually not necessary, though, when, dealing with integers only

→ See "my_equilateral_triangle.m" for an example

Practice question: Parameters and arguments

What would the values of the variables "x" and "y" be after executing the following commands?

```
>> a = 10;
>> b = 20;
>> [x, y] = my_function_ab(b, a)
```

```
1   function [x, y] = my_function_ab(a, b)
2 -   x = a - 2;
3 -   y = b + 2;
4 -   end
```

(A) "x" is 18 and "y" is 12

(B) "x" is 8 and "y" is 22

Practice question: Parameters and arguments

What would the values of the variables "x" and "y" be after executing the following commands?

```
>> a = 10;
>> b = 20;
>> [x, y] = my_function_ab(b, a)
```

```
1   function [x, y] = my_function_ab(a, b)
2 -   x = a - 2;
3 -   y = b + 2;
4 -   end
```

(A) "x" is 18 and "y" is 12

(B) "x" is 8 and "y" is 22

Practice question: Parameters and arguments

What would the values of the variables "x" and "y" be after executing the following commands?

```
>> a = 10;
>> b = 20;
>> [y, x] = my_function_ab(a, b)
```

```
1   function [x, y] = my_function_ab(a, b)
2 -   x = a - 2;
3 -   y = b + 2;
4 -   end
```

(A) "x" is 8 and "y" is 22

(B) "x" is 22 and "y" is 8

Practice question: Parameters and arguments

What would the values of the variables "x" and "y" be after executing the following commands?

```
>> a = 10;
>> b = 20;
>> [y, x] = my_function_ab(a, b)
```

```
1    function [x, y] = my_function_ab(a, b)
2 -    x = a - 2;
3 -    y = b + 2;
4 -    end
```

(A) "x" is 8 and "y" is 22

(B) "x" is 22 and "y" is 8

Function parameters and arguments

```
function [x, y] = my_function_ab(a, b)
x = a - 2;
y = b + 2;
end
```

```
>> who_is_this = 1;
>> i_dont_know = 3;
>> [var1, var2] = my_function_ab(who_is_this, i_dont_know)
    var1 =
        -1
    var2 =
         5
```

From now on, I will use the following words:

Function parameters and arguments

```
function [x, y] = my_function_ab(a, b)
x = a - 2;
y = b + 2;
end
```

```
>> who_is_this = 1;
>> i_dont_know = 3;
>> [var1, var2] = my_function_ab(who_is_this, i_dont_know)
    var1 =
        -1
    var2 =
         5
```

From now on, I will use the following words:

- **Parameters:** the input and output "entities" specified in the functions header
  - Here: a, b, x, and y

Function parameters and arguments

```
function [x, y] = my_function_ab(a, b)
x = a - 2;
y = b + 2;
end
```

```
>> who_is_this = 1;
>> i_dont_know = 3;
>> [var1, var2] = my_function_ab(who_is_this, i_dont_know)
    var1 =
        -1
    var2 =
         5
```

From now on, I will use the following words:

- **Parameters:** the input and output "entities" specified in the functions header
  - Here: a, b, x, and y
- **Arguments:** the data passed to and from a function when calling it
  - Here: `who_is_this`, `i_dont_know`, `var1`, and `var2`

Function parameters and arguments

```
function [x, y] = my_function_ab(a, b)
x = a - 2;
y = b + 2;
end
```

```
>> who_is_this = 1;
>> i_dont_know = 3;
>> [var1, var2] = my_function_ab(who_is_this, i_dont_know)
    var1 =
        -1
    var2 =
         5
```

From now on, I will use the following words:

- **Parameters:** the input and output "entities" specified in the functions header
  - Here: a, b, x, and y
- **Arguments:** the data passed to and from a function when calling it
  - Here: `who_is_this`, `i_dont_know`, `var1`, and `var2`

The names of parameters and arguments do not have to be the same. It is the order that matters

Practice question

Assuming we start with an empty workspace, which of the following statements are true after executing the following commands in the command window?

```
>> array = [1, 4, 6; 3, 2, 5];
>> [a, b, c] = my_function(array);
```

```
1   function [y, w, z] = my_function(x)
2
3 -     y = x(1);
4 -     w = min(x);
5 -     z = x > 2;
6
7 -   end
```

(A) The variable "a" has the value 1

(B) The variable "y" has the value 1

(C) The variable "b" has the value [1, 2, 5]

(D) The variable "c" has the value [0, 1, 1]

Practice question

Assuming we start with an empty workspace, which of the following statements are true after executing the following commands in the command window?

```
>> array = [1, 4, 6; 3, 2, 5];
>> [a, b, c] = my_function(array);
```

```
1   function [y, w, z] = my_function(x)
2
3 -     y = x(1);
4 -     w = min(x);
5 -     z = x > 2;
6
7 -   end
```

(A) The variable "a" has the value 1

(B) The variable "y" has the value 1

(C) The variable "b" has the value [1, 2, 5]

(D) The variable "c" has the value [0, 1, 1]

Practice question

What will the value of variable "a" be after executing the following commands?

```
>> y = @(x) x + 5;
>> a = my_function(y(5));
```

```
1   function [y] = my_function(x)
2
3 -     y = x + 5;
4
5 -   end
```

(A) 5

(B) 10

(C) 15

(D) 20

Practice question

What will the value of variable "a" be after executing the following commands?

```
>> y = @(x) x + 5;
>> a = my_function(y(5));
```

```
1   function [y] = my_function(x)
2
3 -     y = x + 5;
4
5 -   end
```

(A) 5

(B) 10

(C) 15

(D) 20

Practice question

Consider the following array:

```
>> array = [2, 4, 6, 8, 10, 12];
```

Which of the following commands assign the value `10` to the variable "b"?

(A) b = array(4);

(B) b = array(5);

(C) array(4);

(D) array(5);

(E) b = array(end-1);

(F) b = array(1, 5);

(G) b = array(5, 1);

Practice question

Consider the following array:

```
>> array = [2, 4, 6, 8, 10, 12];
```

Which of the following commands assign the value `10` to the variable "b"?

(A) b = array(4);

(B) b = array(5);

(C) array(4);

(D) array(5);

(E) b = array(end-1);

(F) b = array(1, 5);

(G) b = array(5, 1);

Practice question

What will the value of variable "v" be after executing the following commands (note: ... indicates line continuation)?

```
>> array = [2, 1, 6; 8, -1, 0; 4, 0, 1]
     2     1     6
     8    -1     0
     4     0     1
>> v = sum(array(1,:).*array(1:end,end)') / ...
       (array(1,1:end)*array(:,end));
```

(A) 2

(B) 324

(C) 1

(D) NaN

(E) Matlab will throw an error

Practice question

What will the value of variable "v" be after executing the following commands (note: ... indicates line continuation)?

```
>> array = [2, 1, 6; 8, -1, 0; 4, 0, 1]
     2     1     6
     8    -1     0
     4     0     1
>> v = sum(array(1,:).*array(1:end,end)') / ...
       (array(1,1:end)*array(:,end));
```

(A) 2

(B) 324

(C) 1

(D) NaN

(E) Matlab will throw an error