Introduction to Computers for Engineers:

Recitation #2

Learning Objectives

- Learn how to write **functions** in MATLAB
- Understand that functions are templates that can be modified and reused to suit your needs
- Understand that functions are used by calling them
- Understand that you are finished writing a function after you have tested it
- Understand that functions can be used together within other programs to perform larger tasks

Activity 1: Function warm-up

- Write a function with 2 inputs and 2 outputs named math
 - Remember that the name of the function and the filename must both be math
- Inputs:
 - num1: Double
 - num2: Double
- Outputs:
 - difference: num1-num2
 - sum: num1+num2
 - Run your function to make sure it works!
- Questions:
 - How do you call a function?
 - Are there any similarities between the function call and the function header?
 - What does it mean to have an input? Assign an output value? Do we need to print anything or ask the user for anything?

Activity 1: Solutions

```
function [difference, sum] = math(num1, num2)
difference = num1 - num2;
sum = num1 + num2;
end
```

Question: What if you wanted to run the function inside the script?

```
var_1 = 15;
var_2 = 8;
[diff, sum] = math(var_1, var_2);
function [difference, sum] = math(num1, num2)
difference = num1 - num2;
sum = num1 + num2;
end
```

Remember: now the script has to have a different name than the function!

Activity 2: More functions

- Write a function with 2 inputs and 2 outputs named math2
- Inputs:
 - num1: Double
 - num2: Double
- The outputs can be any mathematical operation you want, example:
 - recitation_sucks: (num_1 + num_2) * (num_1^2)
 - im_lost: (num_1*num_2) + (num_1 / num_2)
- In your group:
 - Create test cases (i.e. run it with different inputs to make sure that it works)
 - Discuss: What do you think an appropriate number of test cases is for a program like this?
 - Discuss: Why might functions be useful to write instead of plain scripts?

Activity 2: Solutions

```
Function [recitation_sucks, im_lost] = math2(num_1, num_2)
recitation_sucks = (num_1 + num_2) * (num_1^2);
im_lost = (num_1*num_2) + (num_1 / num_2);
end
```

Activity 3: Calling multiple functions (Example)

Suppose I wanted to add all of the results from the function math and math2



This works as long as all of the MATLAB functions are located in the same directory!

Activity 3: Calling multiple functions

- We are going to create a GPA calculator!
- Let's first create the function totalCredits():
 - total_creds = totalCredits(credit_array)
 - The purpose of function totalCredits() is to compute the total number of credits you've taken. You can use the built-in MATLAB function sum().
- Example:
 - credit_array = [3, 4, 3, 1]
 - total_creds = sum(credit_array)
- Make sure that your function works!

Note: A = 4.0 B+ = 3.5 B = 3.0 C+ = 2.5 C = 2.0 D = 1.0

Activity 3: Calling multiple functions

- We are going to create a GPA calculator!
- Now create the function totalPoints():
 - total_points = totalPoints(credit_array, grade_array)
 - **NOTE:** credit_array and grade_array should align.
- Example:
 - credit_array = [3, 4, 3, 1]
 - grade_array = [4.0, 3.5, 2.5, 0.0]
 - individual_points = credit_array .* grade_array
 - total_points = sum(individual_points)
 - **NOTE:** .* denotes element-wise multiplication of arrays!
- Make sure that your function works!

Note: A = 4.0 B+ = 3.5 B = 3.0 C+ = 2.5 C = 2.0 D = 1.0

Activity 3: Calling multiple functions

- We are going to create a GPA calculator!
- Now create the function computeGPA():
 - gpa = computeGPA(credit_array, grade_array)
 - **NOTE:** credit_array and grade_array should align.
- This function should output the result of dividing the output of totalPoints() by the output of totalCredits()
 - gpa = totalPoints() / totalCredits()
- Make sure that your function works!

Question: Why do you think creating a function to compute the GPA is better than just writing a script? Note: A = 4.0 B+ = 3.5 B = 3.0 C+ = 2.5 C = 2.0 D = 1.0

Activity 3: Solution

```
[] function [total_points] = totalPoints(credit_array, grade_array)
[ function [total_creds] = totalCredits(credit_array)
                                                         individual_points = credit_array .* grade_array;
 total_creds = sum(credit_array);
                                                         total_points = sum(individual_points);
 end
                                                         end
                     [] function [gpa] = computeGPA(credit_array, grade_array)
                      total_creds = totalCredits(credit_array);
                      total_points = totalPoints(credit_array, grade_array);
                      gpa = total_points / total_creds;
                       end
                         >> gpa = computeGPA([3, 4, 3], [4.0, 4.0, 3.5])
                         gpa =
                              3.8500
```

Activity 3: More on functions

- How would you modify the computeGPA() function so that it can compute your GPA for two semesters?
- Hint: Increase your number of input arrays to 4 and re-use your functions!

Activity 4: Make your own sequence of functions!

- Think of your own example where you think leveraging multiple functions would be useful.
- Discuss with your group and implement it together!