Engineering 13300
MATLAB 1: Introduction to MATLAB

NOTE: Use the ***ENGR_MATLAB_Template.m*** provided. For Ma1, add your main code under "Calculations" section and `disp` or `fprintf` under "Outputs" section.

Team Tasks

Recall the guidelines for team activities:

1. You should work as a team; **all** team members will be held responsible for all material. You may work together and contribute to one program and submit similar codes as long as the contributors to the development of the solution are documented.
2. Each student is responsible for submitting their own assignment.

Task 1 (of 7)

Learning Objectives: Perform arithmetic operations in MATLAB (i.e. addition, subtraction, multiplication, division, and exponentiation), accounting for order of operations; Create valid identifiers, accounting for relevant MATLAB rules (e.g. keywords) and code standard.
Computer Operator: Team member with the longest pointer finger

Background: Each programming language has its specific strengths for accomplishing various computational goals by their users. For example, Python's general collection type makes it possible to mix data types in a single collection (lists, tuples, and dictionaries). This makes processing database records easier, such as a personal contact directory containing names, addresses, phone numbers and email. MATLAB, on the other hand, is designed to operate on arrays of numbers and perform numerical processes. This makes it an excellent environment for numerical modeling and analysis. As a result, each environment will be very efficient with some programming actions and very inefficient with others. These exercises explore some of these strengths and weaknesses of various programming languages.

The command window in the MATLAB IDE (Integrated Development Environment) is similar to the Python 3 interpreter environment. They are both interpreters that process lines of code as they are entered. Therefore, you can use the command window to test out specific lines of code. In this exercise, we will go straight to building scripts using the built-in editor in MATLAB, similar to how you generated code with Spyder for Python 3 scripts.

Type **`edit`** in the command window. This launches the MATLAB editor/debugger and creates a new .m file. You can type the same commands in the MATLAB editor as you can in the command window. The only difference is that using the editor, you can run a series of commands as often as you like. For this exercise and all future exercises, unless otherwise indicated, you should use the editor and script files to construct and comment your MATLAB code.

Assuming that equations (1) through (9), shown below, are computed sequentially in the order shown, calculate the result of each expression using:

- Python (save your file as `Ma1_Task1_teamnumber.py`)
- MATLAB (save your file as `Ma1_Task1_teamnumber.m`)

```
1. A = 25
2. B = A ^ 2
3. C = B - A * 7 / 8
4. D = 17 // 3 + 23 / 7
5. E = 4 ^ (5 * 7)
6. F = -4 ^ (6 / 11)
7. M = 4 ^ (i / 2)
8. Z = 179 % 20
9. P = exp(2 * pi)
```

Answer the following questions:

1) What occurs when entering equation 4? What are the differences between Python and MATLAB?
2) Why does equation 6 cause a problem? What happens in MATLAB?
3) In equation 7, how does "`i`" function? Is it possible to use the letter "`i`" as a variable?
4) Why does equation 8 cause a problem in MATLAB? How would you find a command or a function in MATLAB to compute the modulus?
5) How do you calculate the natural logarithm of a value in MATLAB? How do you call the base of the natural logarithm, e, in MATLAB?

Save your response to these questions to `Ma1_team_teamnumber.pdf`.

Task 1 Files:

1) `Ma1_Team_teamnumber.pdf`
2) `Ma1_Task1_teamnumber.py`
3) `Ma1_Task1_teamnumber.m`

Task 2 (of 7)

Learning Objectives: Create and execute simple script comprised of basic MATLAB commands; Output data from a script to the screen in MATLAB; Apply course code standard in development of MATLAB scripts; Comment your code using the percent operator (%) and modularize your code using the (%%) operator in a useful and efficient fashion in MATLAB.
Computer Operator: Team member with the shortest last name

Background: A fuel tank is constructed in the shape of a cylinder. The design engineer that designed the tank provided the dimensions in meters. The construction engineer that is building the tank needs to know the capacity in U.S. gallons, as well as the dimensions in feet.

Part A:
Write a MATLAB script to find the volume of the cylinder in U.S. gallons, as well as the tank dimensions in feet. Assume that the initial measurements are 6 meters in diameter and 10 meters tall.

Comment your code adequately. Start your script by removing any data and output from previous scripts by using `clc` and `clear` prior to performing anything else. Display your final answers to the screen using `disp` and a statement without a semicolon, e.g. write the following into your script

```
disp('The capacity in U.S. gallons is:')

capacity
```

where `capacity` is a variable that you defined in preceding calculations.

***Note: The given values should be assigned at the top of your code and referenced as variables in the body of the code.*

Part B:
In the same MATLAB script, perform the same calculations for a second tank that is 4 meters in diameter and 9 meters tall. This time, use `fprintf` to achieve the following output (no decimals for the capacity and one decimal for the dimensions):

```
The capacity is X U.S. gallons.
The tank has a diameter of X.X ft and is X.X ft tall.
```

Save your script as: `Ma1_Task2_teamnumber.m`

***Note: At this point, creating a flowchart should be done without instructions explicitly asking for it. Append your flowchart to the previously created PDF (Ma1_Team_teamnumber.pdf).*
***Note: we will be covering more on how to use the **fprintf** function in the modules for MATLAB 2.*

Task 2 Files:
1) `Ma1_Task2_teamnumber.m`

Task 3 (of 7)

Learning Objectives: Identify which resources are available to you to aid in learning MATLAB; Access MATLAB "help" in MATLAB and through online MATLAB documentation.
Computer Operator: Team member with the longest hair

Background: MATLAB has a very powerful set of help tools. By going to Help and then going to MATLAB Help, you will open the interactive help files. They operate in a similar fashion to most Windows help files. By clicking on Contents, you will have access to the entire directory of MATLAB help, all commands, and some examples. Clicking on the index tab allows you to search for details on a known function. The Search tab will allow you to search for a function to do a specific task. These two tools can help you find information on any function.

Use the interactive help files to look for more information on the following functions. Test them out in the command window:

- `sin`
- `exp`
- `mod`
- `sqrt`
- `abs`
- `pi`
- semicolons (in relation to output)

MATLAB also has a command line help utility. In the MATLAB command window, you can type `help` *`command`* where command is the item that you want help with. For instance, typing `help sin` will give you information on the sin function. Use the help command to find out more information on the following commands:

- `clc`
- `clear`
- `whos`
- `what`
- `why`
- `ans`

For **all 13** of the above commands, answer the following:

1) What are the arguments required by the function?
2) What is the appropriate syntax of the argument?
3) Do the arguments have any special units?
4) What is the purpose of the function?

Once you have tested all of the commands, answer the following:

1) Describe the difference between the `fprintf` and `disp` commands

Save your answers in your previously created pdf, `Ma1_Team_teamnumber.pdf`.

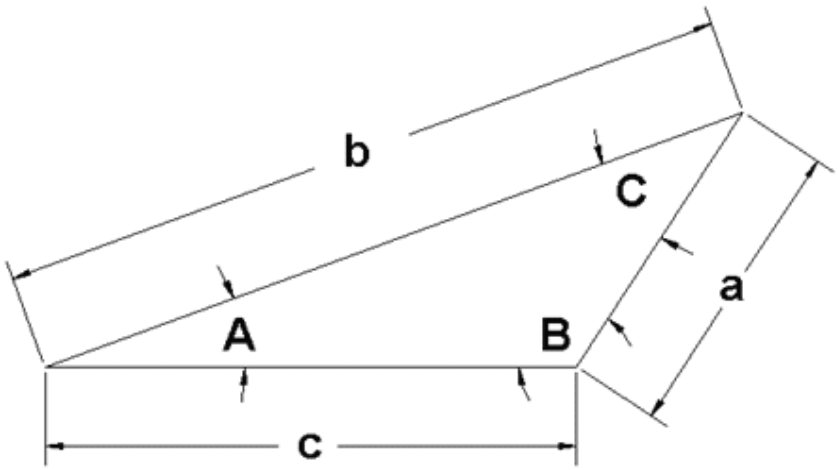
Task 4 (of 7)

Learning Objectives: Create and execute simple functions to perform arithmetic operations in MATLAB (i.e. addition, subtraction, multiplication, division, and exponentiation), accounting for order of operations;

Computer Operator: Team member who slept the least last night

Given 1 side and 2 angles of an oblique triangle (see figure), determine the unknown distances, a and c, and the unknown angle, B. The function will print all values to the screen. Make sure to add sufficient comments.

Distance b: 1300 meters
Angle A: 25 degrees
Angle C: 37 degrees



***Note: Triangle shown is not to scale.*

The program should output the lengths a and c to the nearest meter and the angle B to the nearest tenth of a degree using `fprintf`.

Example output (command line input in bold):

```
Distance b: 1300
Angle A: 25
Angle C: 37
Distance a: 622 meters
Distance c: 886 meters
Angle B: 118.0 degrees
>>
```

Task 4 Files:

1) `Ma1_Task4_teamnumber.m`

Task 5 (of 7)

Recall the guidelines for activities:

1. You should work as a team; **all** team members will be held responsible for all material.
2. You should work on this Task using one computer for the entire team unless otherwise directed.
3. If you are not making progress, take action to get unstuck.
4. Do not write on the activity sheets and be sure to return them at the end of class.

Learning Objectives: Practice plotting data using MATLAB. Use colon operator to perform operations across values in an array.

Computer Operator: The person whose birthday is the closest to Flag Day. (Look up the date if you don't know!)

Background:

The sine function is often utilized in engineering when making calculations to evaluate systems. Many approximations of the sine function have been found. One of these is the Taylor series approximation, which is indicated by the following equation, where y approximates the value of sine and x is equal to the angle in radians:

$$y = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!}$$

Using the colon and the dot operators, create a vector called X with values from -50 to 50 in increments of 0.1 and compute the values of y in a new vector called Y. Plot the values of Y with respect to the values of X. Using the `xlabel()`, `ylabel()`, and `title()` functions, label the graph appropriately. Experiment with changing the color of the line to blue and the markers to squares.

***Note: You can use `clc` and `clear` command to remove any previously generated data/output such that your program is off to a fresh/clean start.*

Save your script as:

`Ma1_Task5_teamnumber.m`

Task 5 Files:

1) `Ma1_Task5_teamnumber.m`

Individual Tasks

Guidelines for Tasks 6–7:
Tasks 6–7 are individual tasks. You may seek help from classmates, the instructional team or others but the work you submit should be your own. If you collaborate with others and use information developed together or by someone else, ALWAYS document and reference that material.

Task 6 (of 7):

Learning Objectives: The following exercise highlights the usage of MATLAB as a Calculator

Part A:

Use MATLAB to calculate the value of each expression. Record the MATLAB command and the result of each of the variables on the ***answer sheet***.

$$p = (2+7)^3 + \frac{273^{2/3}}{2} + \frac{55^2}{3}$$

$$q = 2^3 + 7^3 + \frac{273^2}{2} + 55^{\frac{2}{3}}$$

$$r = \left|1 - 0.4 \cdot \tan^{-1}\left(\frac{\pi}{6}\right)\right|$$

Part B:

Define `x` and `z` as `x = 9.6` and `z = 8.1`. Then evaluate each expression. Record the MATLAB command and result of each variable on the answer sheet.

$$a = xz^2 - \left(\frac{2z}{3x}\right)^{3/5}$$

$$b = \frac{443z}{2x^3} + \frac{e^{-xz}}{x+z}$$

$$c = \ln(z)$$

$$d = \log(z)$$

Hint: "$\ln(\cdot)$" or "$\log_e(\cdot)$" is commonly known as the natural logarithm and "$\log(\cdot)$" or "$\log_{10}(\cdot)$" is commonly known as log to the base 10.

Task 6 Files:

1. `Ma1_Ind_username.pdf (answer sheet)`

Task 7 (of 7):

Learning Objectives: The following exercise demonstrates matrix manipulations

Part A:

In the script file, create the matrix shown:

$$Amatrix = \begin{bmatrix} 2 & 5 & 8 & 5 \\ 10 & 9 & 1 & 4 \\ 6 & 3 & 2 & 10 \end{bmatrix}$$

To create a matrix, you will need to create multiple rows of vectors. For example, to create matrix

$$x = \begin{bmatrix} 1 & 3 \\ 9 & 2 \end{bmatrix}$$

type `x = [1,3;9,2]` or `x = [1 3;9 2]`. Note that semicolons separate each row in the matrix. You can also use "enter" (a new line) to start a new row.

Now type each of the command in the table below in your script, run the script, and find out what each of the commands do. Write comments for each of the lines describing the function of the commands.

Command
`Bvector = Amatrix(1,:)`
`Cvector = Amatrix(2,:)`
`Dvector = Amatrix(:,3)`
`sort(Dvector)`
`Amatrix(3)= 30; Amatrix`
`Evector = linspace(1,25,4)`
`Fvector = Evector*5`
`Amatrix(1:2)`
`Amatrix(2:3)`
`Amatrix(1:2,2:3)`

***Notes:*

- *If you need help completing this Problem, use the **MATLAB help function** to look up the terms `colon`, `sort`, and `linspace`.*
- *You do not need to submit these commands in the answersheet or the `.m` file with these commands; Task 7 Part A is intended to prepare you for Task 7 Part B.*

Part B:

Write a MATLAB command for each function in the table below. Record the command you used on the answer sheet:

Function
Create **`Bmatrix`** with values identical to **`Amatrix`** except the middle row is **`Bvector`**.
Create **`Gvector`** by extracting the third row in **`Amatrix`**.
Extract element in row 2, column 3 from **`Amatrix`**
Replace the value 2 in **`Amatrix`** (row 1 and column 1) with the value 55.

Task 7 Files (same file as Task 6, keep working in this file):

1. `Ma1_Ind_username.pdf`