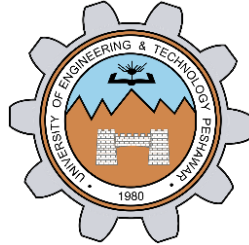


MATLAB TRAINING

LAB # 01



Fall 2023

CSE-402L


Digital Signal Processing Lab

Submitted by: **AIMAL KHAN**

Registration No.: **21PWCSE1996**

Class Section: **A**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Student Signature: 

Submitted to:

Dr. Yasir Saleem Afridi.

Wednesday, September 27, 2023

Department of Computer Systems Engineering
University of Engineering and Technology, Peshawar

CSE 402L: Digital Signals Processing

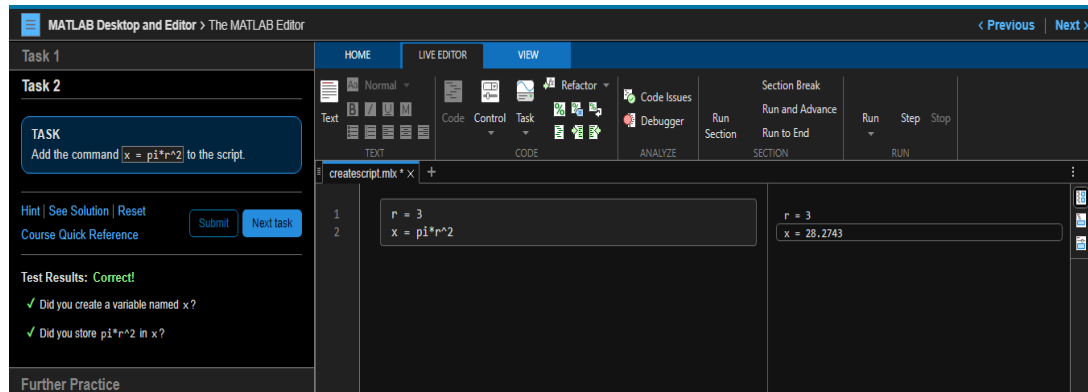
	Poor (Does not meet expectation (1))	Fair (Meet Expectation (2-3))	Good (Exceeds Expectation (4-5))	Score
Demonstration of Concepts	The student failed to demonstrate a clear understanding of the assignment concepts	The student demonstrated a clear understanding of some of the assignment concepts	The student demonstrated a clear understanding of the assignment concepts	30%
Accuracy	The student mis-configured enough signal processing settings that the computer couldn't function properly.	The student configured enough signal processing settings that the computer partially functioned	The student configured the signal processing settings that the computer fully functioned	30%
Following Directions	The student clearly failed to follow the verbal and written instructions to successfully complete the lab	The student failed to follow the some of the verbal and written instructions to successfully complete all requirements of the lab	The student followed the verbal and written instructions to successfully complete requirements of the lab	20%
Time Utilization	The student failed to complete even part of the lab in the allotted amount of time	The student failed to complete the entire lab in the allotted amount of time	The student completed the lab in its entirety in the allotted amount of time	20%

MATLAB Training

Tasks:

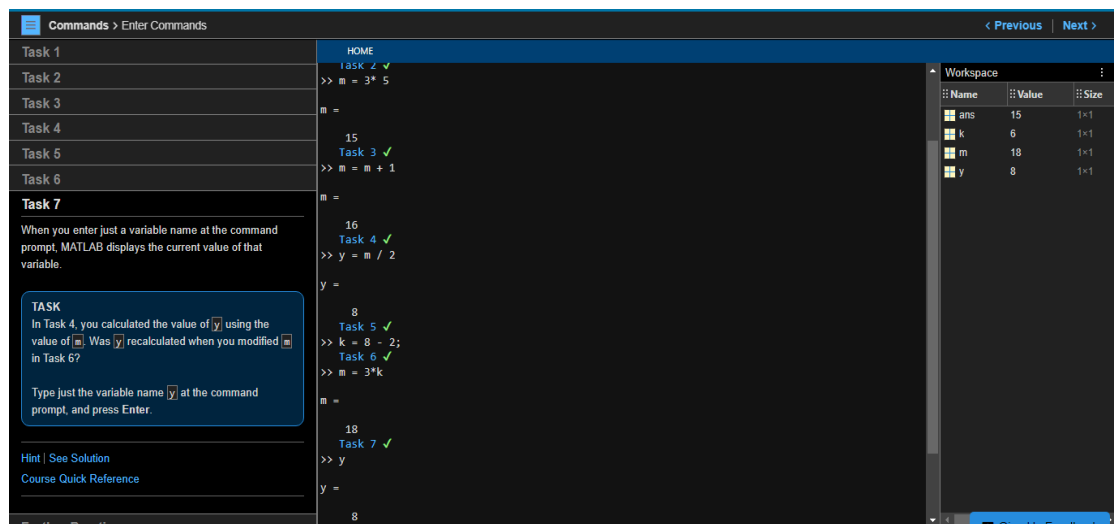
➤ Course Overview

- Objective: Familiarize yourself with the course.
- Remarks along with final snapshot



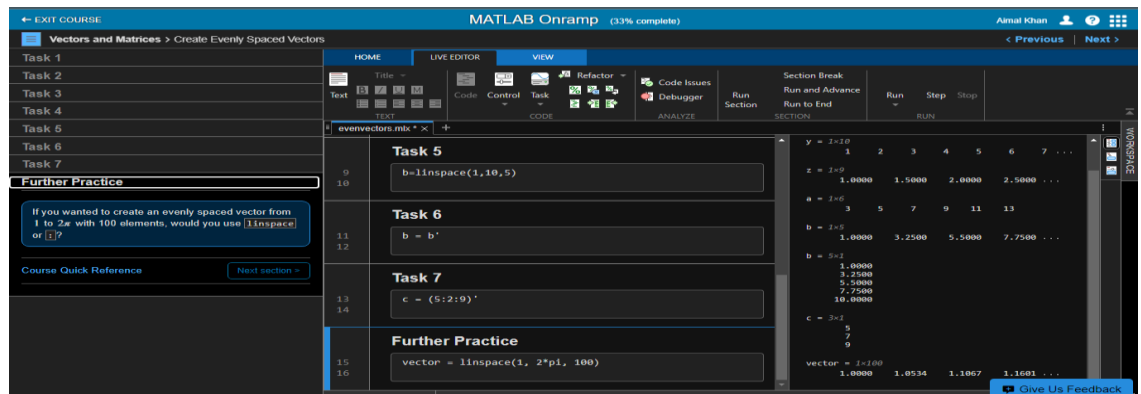
➤ Commands

- Objective: Enter commands in MATLAB to perform calculations and create variables.
- Remarks along with final snapshot



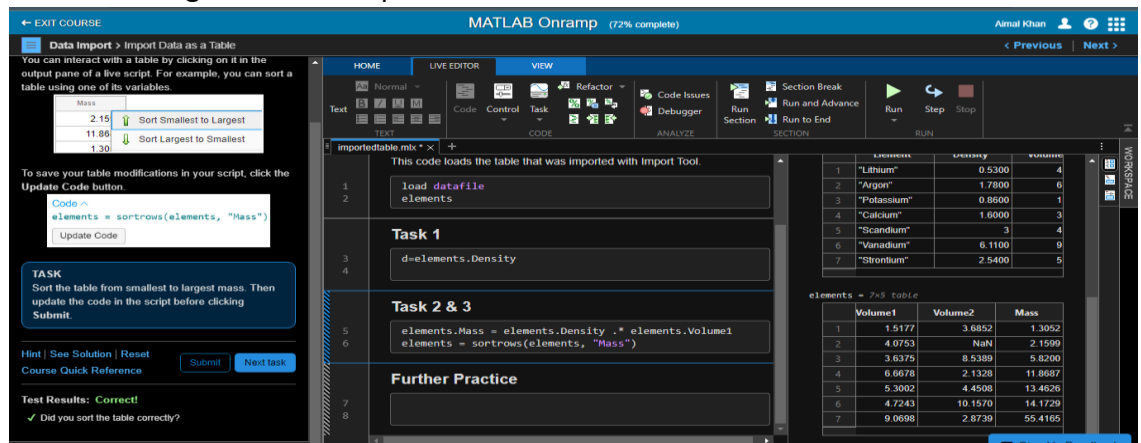
➤ Vectors and Matrices

- Objective: Create MATLAB variables that contain multiple elements.
- Remarks along with final snapshot



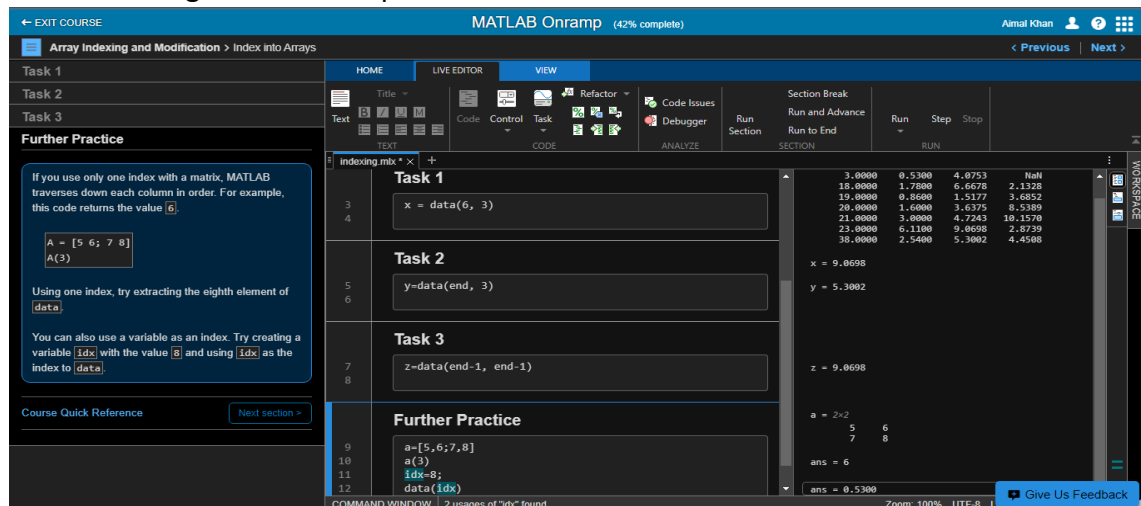
➤ Importing Data

- Objective: Bring data from external files into MATLAB.
- Remarks along with final snapshot



➤ Indexing into Modifying Arrays

- Use indexing to extract and modify rows, columns, and elements of MATLAB arrays.
- Remarks along with final snapshot



➤ Array Calculations

- Objective: Perform calculations on entire arrays at once.
- Remarks along with final snapshot

The screenshot shows the MATLAB Onramp interface for a task titled "Request Multiple Outputs in Function Calls". The task pane on the left describes "Task 3": finding the maximum value of a vector and its corresponding index using the `max` function. It includes a hint, a "Submit" button, and test results showing "Correct!". The code editor shows the following code:

```
Task 1
4 dsize = size(data)
5

Task 2
6 [dr, dc] = size(data)
7

Task 3
8 [vMax, ivMax] = max(v2)
9

Further Practice
10
11
```

The workspace on the right shows the following variables:

```
v2 = 7x1
    0.5000
    2.1328
    3.6852
    8.5389
   10.1570
    2.8739
    4.4588

dsize = 1x2
     7     4

dr = 7
dc = 4
vMax = 10.1570
ivMax = 5
```

➤ Calling Functions

- Objective: Call functions to obtain multiple outputs
- Remarks along with final snapshot

The screenshot shows the MATLAB Onramp interface for a task titled "Create Arrays with Functions". The task pane on the left describes "Task 3": creating a matrix of all zeros that has 6 rows and 3 columns (6-by-3) by using the `zeros` function. It includes a hint, a "Submit" button, and test results showing "Correct!". The code editor shows the following code:

```
Task 1
1 x = rand(5)
2

Task 2
3 y = rand(5, 1)
4

Task 3
5 z = zeros(6, 3)
6
```

The workspace on the right shows the following variables:

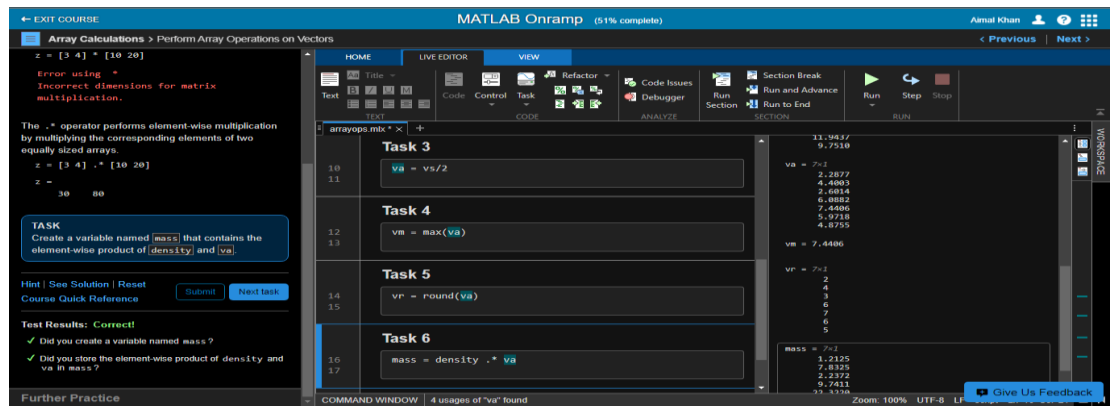
```
x = 5x5
    0.8147    0.0975    0.1576    0.1419 ...
    0.9858    0.2785    0.9706    0.4218
    0.1270    0.5469    0.9572    0.9157
    0.9134    0.9575    0.4854    0.7922
    0.6324    0.9649    0.8003    0.9595

y = 5x1
    0.7577
    0.7431
    0.3922
    0.6555
    0.1712

z = 6x3
     0     0     0
     0     0     0
     0     0     0
     0     0     0
     0     0     0
     0     0     0
```

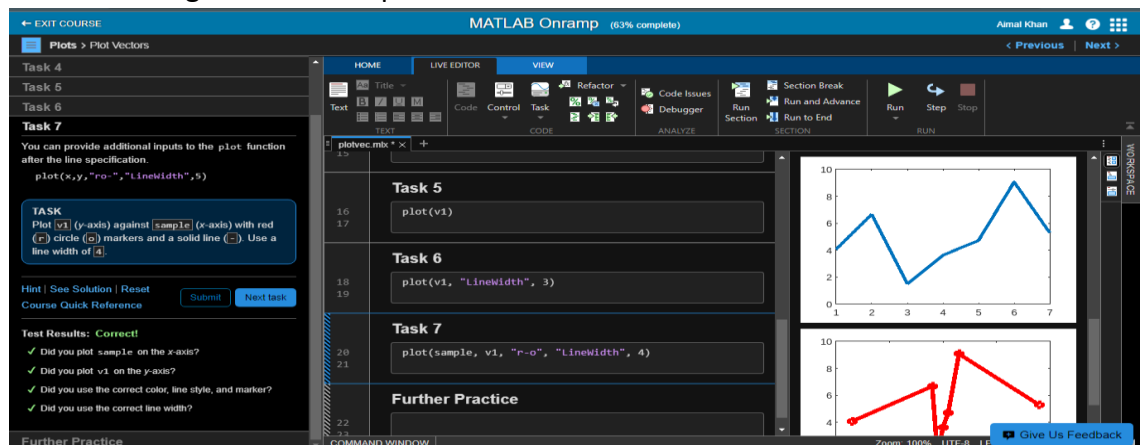
➤ Obtaining Help

- Objective: Use the MATLAB documentation to discover information about MATLAB features.
- Remarks along with final snapshot



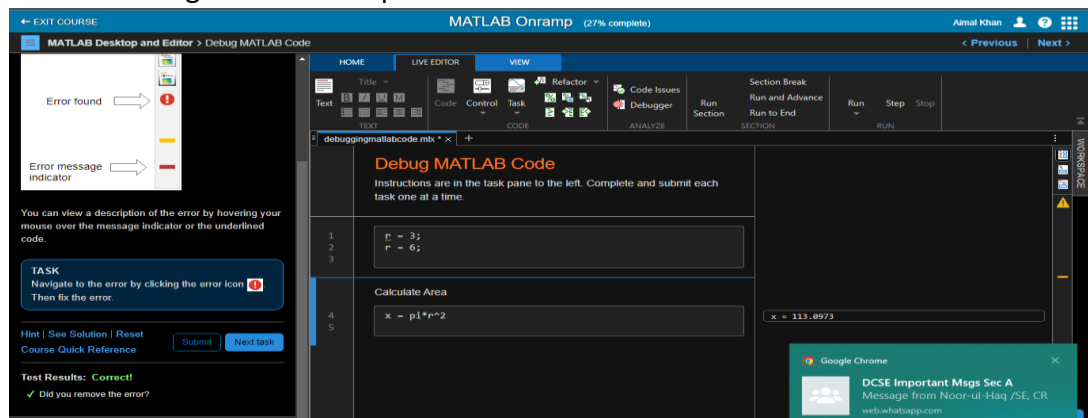
➤ Plotting Data

- Visualize variables using MATLAB's plotting functions.
- Remarks along with final snapshot



➤ Review Problems

- Objective: Bring together concepts that you have learned with a project.
- Remarks along with final snapshot



➤ MATLAB Scripts

- Objective: Write and save your own MATLAB programs
- Remarks along with final snapshot

The screenshot shows the MATLAB Onramp interface at 45% completion. The current task is 'Array Indexing and Modification > Extract Multiple Elements'. The interface includes a task list on the left, a central code editor with tabs for HOME, LIVE EDITOR, and VIEW, and a workspace on the right. The task instructions for Task 4 state: 'You can use a single range of index values to reference multiple vector elements. For example, this code returns a subset of vector `v` containing the elements from index 3 to the end. `x = v(3:end)`'. The task objective is to 'Create a vector named `q` containing the 2nd through 5th elements of `density`'. The code editor shows the following code:

```
density = data(:, 2);  
  
volumes = data(:, end-1:end);  
  
p = density(6);  
  
q = density(2:5, :);
```

 The workspace on the right displays the following variables: `density` (7x2 matrix), `volumes` (7x2 matrix), `p` (6.1100), and `q` (4x1 vector). The test results show 'Correct!' and two green checkmarks indicating successful completion of the task.

➤ Logical Arrays

- Objective: Use logical expressions to help you to extract elements of interest from MATLAB arrays
- Remarks along with final snapshot

The screenshot shows the MATLAB Onramp interface at 75% completion. The current task is 'Logical Arrays > Logical Indexing'. The interface includes a task list on the left, a central code editor with tabs for HOME, LIVE EDITOR, and VIEW, and a workspace on the right. The task instructions for Task 5 state: 'You can use logical indexing to reassign values in an array. For example, to replace all values in the array `x` that are equal to 999 with the value 1, use this syntax. `x(x==999) = 1`'. The task objective is to 'Modify `v1` so that any value less than 4 is replaced with the value 0'. The code editor shows the following code:

```
y = v1 < 4;  
z = v1(v1 < 4);  
a = sample(v1 < 4);  
v1(v1 < 4) = 0;
```

 The workspace on the right displays the following variables: `y` (7x1 logical array), `z` (2x1 vector), `a` (2x1 vector), and `v1` (7x1 vector). The test results show 'Correct!' and two green checkmarks indicating successful completion of the task.

➤ Programming

- Objective: Write programs that execute code based upon some condition.
- Remarks along with final snapshot

The screenshot shows the MATLAB Onramp interface at 81% completion. The task is titled "Task 2" and involves decision branching. The code in the editor is as follows:

```
x = rand;
if x > 0.5
    y = 3;
else
    y = 4;
end
```

The task description states: "To execute some other code if the condition is not met, you can use the `else` keyword." A hint box suggests modifying the script to display the density of an element when the condition is not satisfied. The test results show that the code is correct for both conditions.

The workspace shows a variable `doPlot` with a value of 1. The command window shows the execution of the code, including the loading of a datafile and the plotting of sample densities. The plot is titled "Sample Densities" and shows the density of various elements (Lithium, Argon, Potassium, Calcium, Scandium, Vanadium, Strontium) in g/cm³.

➤ Final Project

- Objective: Bring together concepts that you have learned with a project
- Remarks along with final snapshot

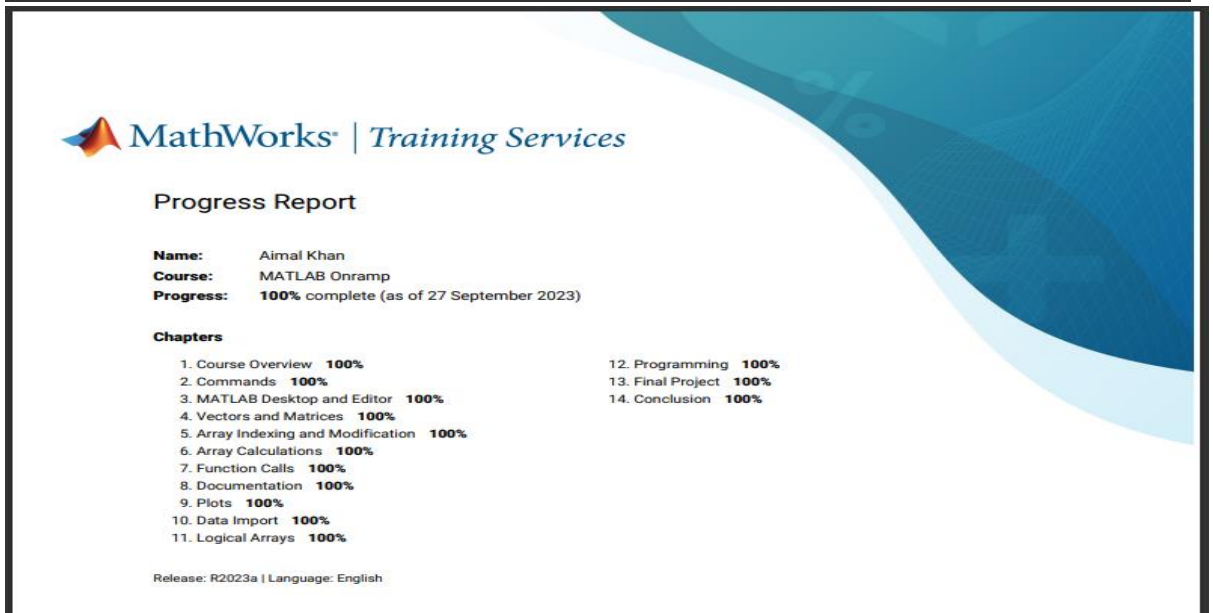
The screenshot shows the MATLAB Onramp interface at 90% completion. The task is titled "Task 6" and involves comparing stellar spectra. The code in the editor is as follows:

```
for v = 1:7
    s = spectra(:, v);
    if speed(v) <= 0
        plot(lambda, s, "--");
    else
        plot(lambda, s, "LineWidth", 3);
    end
    hold on;
end
hold off;
legend(starnames);
```

The task description states: "In the plot, you identify stars with redshifted spectra by using their line styles, and then look up their names in the legend. Can you determine the names of the redshifted spectra without a `for` loop?" A hint box suggests creating a variable `movaway` that contains the elements in `starnames` corresponding to where `speed` is greater than 0. The test results show that the code is correct for all three conditions.

The workspace shows a variable `movaway` with a value of `3x1 string`. The command window shows the execution of the code, including the plotting of stellar spectra and the identification of stars with redshifted spectra. The plot is titled "starplots.mlx" and shows the spectra of various stars (HD 30584, HD 10032, HD 64191, HD 5211, HD 56030, HD 94028, SAO102986) with their corresponding line styles and colors.

➤ Certificates



Reference:

<https://matlabacademy.mathworks.com/progress/share/certificate.html?id=7d515b56-6bf1-484c-9087-3f7d917f8f62&>

<https://matlabacademy.mathworks.com/progress/share/report.html?id=7d515b56-6bf1-484c-9087-3f7d917f8f62&>

The End.