Introduction to MATLAB

LAB 1

ENGG100 - Spring 2024

Consultation

Tuesdays, 14:30 - 15:30

3rd Floor, Engineering Faculty Office

nnadeem@uow.edu.au

Marking Scheme

Attendance is mandatory for all labs

Late: -2 marks

Lab Tasks: 7 marks

Bonus Task: 3 marks

Total Marks per Lab: **10 marks**

IT Support

Office 3.55, 3rd Floor

Timings:

- 8 am - 8 pm, Monday to Friday
- 8 am - 2 pm, Saturday - Sunday

Email:

technicalsupport@uowdubai.ac.ae

Contact:

+971 4 278 1880

Lab Submissions

- Submission links open during lab time on Moodle
- Lab report must be submitted before the next lab session
- All lab code must be compiled into a Word document, with questions labelled, including the bonus task
- Documents must include screenshots of results
- Marks for the lab will be posted the following day on Moodle

Computer Lab Rules

- No eating or drinking inside the labs
- Leave your desk clean and ready for the next student to use
- Remember to log OUT once you have finished
- Avoid saving files on the desktop, keep your files on OneDrive

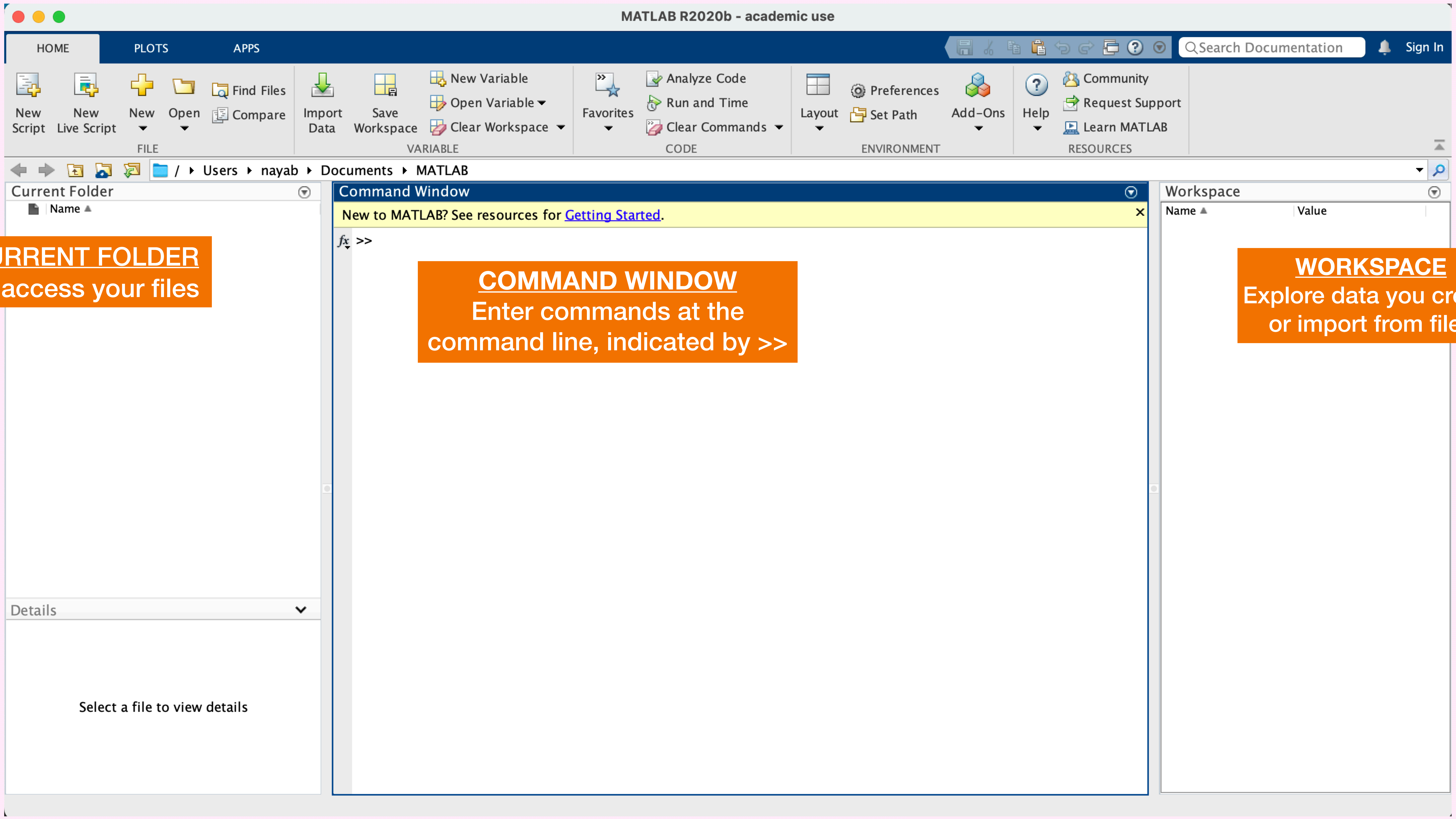
Installing MATLAB

- Installation guide for MATLAB is available on Moodle
- Preferred version - 2022b
- Additional documentation can be accessed using the `doc` command
- Additional help regarding functions can be accessed using the `help` command

Lab 1 Objectives

- Perform arithmetic operations in the command window
- Assign values to variables and use variables for arithmetic operations
- Use built-in functions/variables for arithmetic operations
- Write a basic script which asks the user for some information and returns an answer

MATLAB Layout



New to MATLAB? See resources for Getting Started.

```
>>
```

Select a file to view details

CURRENT FOLDER
To access your files

COMMAND WINDOW
Enter commands at the command line, indicated by >>

WORKSPACE
Explore data you create or import from files

Handy Hints

- When MATLAB is ready to receive a new command, >> is displayed
- If you do not see the >> sign, it may mean MATLAB is busy or waiting for more code before executing the command
- If MATLAB is stuck processing, you can use CTRL + C to exit the loop
- You can use the up arrow key to write the same line of code again
- If you end a command with a semicolon (;), the answer to the line of code will not be displayed after the code is executed
- Use `clear` to clear all your variables (workspace) and `clc` to clear your command window

Perform Arithmetic Operations

+	Addition
-	Subtraction
/ or \	Division (Invert second number & Multiply) or Invert first number * Multiply
*	Multiply
^	Power

Assigning Values to Variables

```
miles_to_km = 1.61;
```

- This will store the value `1.61` into a variable called `miles_to_km`
- The = sign will store whatever is on the right hand side to the left hand side
- To find out how many kilometres there are in 5 miles, we will run the command:

```
miles_to_km * 5
```

<u>Note</u>: Values in variables can be overwritten

TASK 1

- A car is traveling 10.2 miles/hour. Convert this speed to kilometers/hour.
- Formula:

KPH = MPH x 1.6

Using built-in functions and variables

- Built-in variable example: `pi`
- Built-in function examples:

cos(), sin(), tan()	Calculates the cosine, sine & tangent of a number
sqrt()	Calculates the square root of a number
exp()	Calculates the exponential of a number
num2str()	Converts a number type to a string type variable
disp()	Displays an array/text
input()	Prompts the user for an input

TASK 2

- Multiply your student number with pi, find the square root, and assign it to a variable `student_id`

Strings & Numbers

- Square brackets are used to concatenate strings and numbers in MATLAB

```
a = [4,7,0]

b = [a,20]

c = [‘join this string’, ‘ with this string’]

d = [‘this will be strange’, a]

e = [‘but this will work’, num2str(a)]
```

Basic Script

- Programs generally require interactions with humans to gain data
- We will write a basic script that will ask the user their age and return the year they were born in. To do this we will:

1. Write a message to the user requesting data
2. Store the data in a variable
3. Calculate the year they were born
4. Write a message in the command window informing the user the year they were born

- For multiple lines of code, we will be creating a script

Task 3

- Create a new script called “Lab1_Task3_StudentID”
- Create a variable called `age` and use the input function to prompt the user to enter their age
  - Create a variable called `birth_year` and calculate the user’s age using current year and the variable `age`
  - Use the `disp()` function to display the birth year to the user. Remember to convert the variable `birth_year` from a number to a string variable
  - Once the script is saved, click “Run” and verify your script works as expected

Task 4 (Bonus Task)

- Create a new script called “Lab1_Task4_StudentID”
- In this script, ask the user for their height and weight and return their Body Mass Index (BMI).
- BMI = mass / (height ^ 2)
- Height is in m and weight is in kg

Lab Submission

- Ensure your report has your name & student ID in the header
- Ensure all the tasks have been labelled with the code & screenshot of results
- You can submit your report as a DOC or PDF
- Moodle Submissions will close once the lab ends, but you may be able to submit 15 minutes after the lab
- Ensure your files are saved in OneDrive, not locally on the PC
- Ensure to leave your workstation clean and sign out before leaving the lab