**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **06/07/2020** | **Name:** | **Priya P Rao** |
| **Course:** | **MATLAB Onramp** | **USN:** | **4AL18EC041** |
| **Topic:** | * **Course Overview** * **Commands** * **MATLAB Desktop and Editor** * **Vectors and Matrices** | **Semester & Section:** | **4TH sem ‘A’ section.** |
| **Github Repository:** | **Priya-Rao** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session**  **C:\Users\Pawan\Desktop\m1.PNG**  **C:\Users\Pawan\Desktop\m3.PNG** |
| **In today’s session I have learnt about:**   * **Chapter 1: Course Overview**   **MATLAB Onramp is an online course that provides a brief introduction to the MATLAB language. The course gives us hands-on MATLAB experience via the use of an integrated, web-based version of MATLAB**   * **Chapter 2: Commands** * **Entering Commands :** * **We can execute commands by entering them in the command window after the MATLAB prompt (>>) and pressing the Enter Key.**   **To multiply the numbers 2 and 3 : 2\*3**   * **Unless otherwise specified, MATLAB stores calculations in a variable named ans.** * **The Equals sign in the MATLAB is the assignment operator, meaning that the expression on the right of the equal sign is assigned to the variable on the left.**   **Eg.: When we enter x=3+4, MATLAB first evaluates 3+4 and then assigns the results (7) to the variable x.**   * **Adding a semicolon (;) to the end of the command will suppress the output, though the command will still be executed. When we enter a command without semicolon at the end, MATLAB displays the result in the command prompt.**   **We can recall previous commands by pressing the Up arrow key on keyboard.**   * **Naming Variable :**   **We can name our MATLAB variables anything. They should start with a letter and contain only letters, numbers and underscore.**     * **Saving and Loading Variables :** * **We can save variables in our workspace to a MATLAB specific file format called a MAT – file using the save command.**   **To save the work space to a MAT-file named**  **filename.mat, use the command**  **>> save filename**   * **To remove all the variables from workspace, command used is clear function.** * **We can load variables from a MAT-file using the load command.**   **>> load filename**   * **To see contents of any variable, enter the name of the variable.** * **We can use clc command to clean up the Command Window.** * **Using Built-in Functions and Constants :** * **MATLAB contains built – in constants.**   **Eg.: pi = 3.14**   * **MATLAB contains wide variety of built – in functions such as abs(absolute value) and eig(calculate Eigen values).** * **Chapter 3: MATLAB Desktop and Editor** * **MatLab Desktop and Editor :** * **The MatLab Editor :** * **We can enter commands in a script by clicking on the gray code box. And when we are ready, we can submit code.** * **Running Scripts :** * **This live scripts contains formatted text, code and section breaks. In this course, scripts will include task headers to show where we should enter our codes. We can test code before submitting.** * **Chapter 4: Vectors and Matrices** * **Manually Entering Arrays :** * **We can create arrays with multiple elements using square brackets.** * **When we separate numbers by space or comma, MATLAB combines the numbers in a Row. When we separate numbers by semicolons, MATLAB combines the numbers in column.** * **We can combine spaces and semicolons to create a matrix, which is an array with multiple rows and columns. When entering a matrix, we must enter them row by row.** * **In MATLAB, we can perform calculations within the square brackets.**      * **Creating Evenly – Spaced vectors :** * **Shorthand method for creating evenly – spaced vectors is to use the : operator and specify only the start and end points.** * **The : operator uses a default spacing of 1, however we specify our own spacing by,**   **x = 4:2:10**  **x = 4 6 8 10**   * **If we know the number of elements we want in a vector, we can use the linspace function.**   **linspace(first, last, number\_of\_elements)**   * **Both linspace and the : operator create row vectors. We can convert a row vector into a column vector using the transpose operator (‘).**      * **Array Creation Functions :** * **MATLAB contains many functions that help us to create commonly used matrices, such as matrices of random numbers.**   **Eg.: x = rand(2)**   * **To create a matrix of all zeros, use zero function.**   **Eg.: x = zero(5)** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **06/07/2020** | **Name:** | **Priya P Rao** |
| **Course:** | **Internet of Things (IoT)** | **USN:** | **4AL18EC041** |
| **Topic:** | **Introduction to IoT** | **Semester & Section:** | **4TH sem ‘A’ section.** |
| **Github Repository:** | **Priya-Rao** |  |  |

|  |
| --- |
| **AFTERNOON SESSION DETAILS** |
| **Image of session**  **C:\Users\Pawan\Desktop\IOT1.PNG**  **C:\Users\Pawan\Desktop\IOT2.PNG** |
| **In today’s session I have learnt about:**   * **How the current digital transformation is creating unprecedented economic opportunity.** * **How the IoT is bridging the gap between operational and informational technology systems.** * **Discover how standard business processes are being transformed.** * **The security concerns that must be considered when implementing IoT solutions.** |