**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **08/07/2020** | **Name:** | **PRIYA P RAO** |
| **Course:** | **MATLAB Onramp** | **USN:** | **4AL18EC041** |
| **Topic:** | * **Calling Function** * **Obtaining Help** * **Plotting Data** | **Semester & Section:** | **4th sem ‘A’ section.** |
| **Github Repository:** | **Priya-Rao** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session**  **C:\Users\Pawan\Desktop\a1.PNG**  **C:\Users\Pawan\Desktop\a2.PNG**  **C:\Users\Pawan\Desktop\a3.PNG** |
| **In today’s session I have learnt about**   * **Chapter 1: Calling Function :**   **Call functions to obtain multiple inputs.**   * **Obtaining Multiple Outputs from Function Calls :** * **The size function can be applied to an array to produce a single output variable containing the array size. s = size(x)** * **The size function can be applied to a matrix to produce either a single output variable or two output variables. Use square brackets ([ ]) to obtain more than one output. [xrow,xcol] = size(x)** * **The maximum value of a vector and its corresponding index value can be determined using the max function. The first output from the max function is the maximum value of the input vector. When called with two outputs, the second output is the index value. [xMax,idx] = max(x)** * **Chapter 2: Obtaining Help**   **Used the MATLAB documentation to discover information about MATLAB features.**   * **Obtaining Help :** * **The MATLAB documentation contains examples and information that can help you when working on your own problems.** * **Chapter 3: Plotting Data**   **Visualize variables using MATLAB’s plotting functions.**   * **Plotting Vectors :** * **Two vectors of the same length can be plotted against each other using the plot function. plot(x,y)** * **The plot function accepts an additional argument that allows we to specify the color, line style, and marker style using different symbols in single quotes. plot(x,y,"r--o")** * **Notice that each plot command created a separate plot. To plot one line on top of another, use the hold on command to hold the previous plot while you add another line.** * **While the hold state is on, plots will continue to go on the same axes. To return to the default plot behavior, where each plot gets its own axes, enter hold off.** * **When we plot a single vector by itself, MATLAB uses the vector values as the *y-axis* data and sets the *x-axis* data to range from 1 to n (the number of elements in the vector).** * **The plot function accepts optional additional inputs consisting of a property name and an associated value. plot(y,"LineWidth",5)** * **The command above plots a heavy line. We can learn more about available properties in the documentation for**[**Line Properties**](https://www.mathworks.com/help/matlab/ref/matlab.graphics.chart.primitive.line-properties.html)**.** * **We can provide additional inputs to the plot function after the line specifier. plot(x,y,"ro-","LineWidth",5)** * **The command above plots a red (r) dashed (--) line with a circle (o) as a marker. We can learn more about the symbols available in the documentation for**[**Line Specification**](http://www.mathworks.com/help/matlab/ref/linespec.html)**.** * **Annotating Plots :** * **Labels can be added to plots using plot annotation functions, such as title. The input to these functions is a string. Strings in MATLAB are enclosed in double quotes ("). title("Plot Title")** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **08/07/2020** | **Name:** | **PRIYA P RAO** |
| **Course:** | **Internet of Things (IoT)** | **USN:** | **4AL18EC041** |
| **Topic:** | **Everything Becomes Programmable** | **Semester & Section:** | **4th sem ‘A’ section** |
| **Github Repository:** | **Priya-Rao** |  |  |

|  |
| --- |
| **AFTERNOON SESSION DETAILS** |
| **Image of session**  **C:\Users\Pawan\Desktop\a4.PNG**  **C:\Users\Pawan\Desktop\a5.PNG** |
| **In today’s session I have studied about :**   * **Chapter 1: Everything Becomes Programmable** * **Introduction** * **Apply Basic Programming to Support IoT Devices** * **Basic Programming Concepts** * **Basic Programming Using Blockly** * **Programming with Python** * **Prototyping Your Idea** * **What is Prototyping?** * **Prototyping Resources**   **Summary :**   * **This chapter began by discussing how to apply basic programming to support IoT devices. Flowcharts are diagrams that are used to represent processes. There are two common types of computer software: system software and application software. Application software programs are created to accomplish a certain task. System software works between the computer hardware and the application program. Programming variables can be classified into two categories:** * **Local Variables - These are variables that are within the scope of a program.** * **Global Variables - These are variables that are in the scope for the time of the program’s execution. They can be retrieved by any part of the program.** * **The most common logic structures are IF – THEN, FOR Loops, and WHILE Loops.** * **Blockly is a visual programming tool created to help beginners understand the concepts of programming. Blockly implements visual programming by assigning different programming structures to colored blocks.** * **Python is a very popular language that is designed to be easy to read and write. Python is an interpreted language; therefore, an interpreter is required to parse and execute Python code. Variables are labeled memory areas that are used to store runtime program data. Python supports many useful functions and datatypes including Range(), Tuples, Lists, Sets, Dictionary. Python also implements two sub-structures named ELSE and ELIF.** * **Next, the chapter detailed prototyping. Prototyping is the process of creating a rudimentary working model of a product or system. A team at Google used the “Rapid Prototyping Method” to create the Google Glass. The Internet allows for idea exchanges on a whole new level.** |