**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **8/July/2020** | **Name:** | **Prashantha naik** |
| **Course:** | **MATLAB Onramp** | **USN:** | **4al17ec074** |
| **Topic:** | **Review Problems,** i**mporting Data,** **Logical Arrays,** **Programming,** **Final Project,** **Conclusion** | **Semester & Section:** | **6th b** |
| **GitHub Repository:** | **prashanth\_course** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session** |
| **Report – Report can be typed or hand written for up to two pages.**  **Array Comparison with Relational Operators**  Relational operators compare operands quantitatively, using operators like “less than”, “greater than”, and “not equal to.” The result of a relational comparison is a logical array indicating the locations where the relation is true.  These are the relational operators in MATLAB®.   | **Symbol** | **Function Equivalent** | **Description** | | --- | --- | --- | | < | lt | Less than | | <= | le | Less than or equal to | | > | gt | Greater than | | >= | ge | Greater than or equal to | | == | eq | Equal to | | ~= | ne | Not equal to |  Array ComparisonNumeric Arrays The relational operators perform element-wise comparisons between two arrays. The arrays must have compatible sizes to facilitate the operation. Arrays with compatible sizes are implicitly expanded to be the same size during execution of the calculation. In the simplest cases, the two operands are arrays of the same size, or one is a scalar. For more information, see [Compatible Array Sizes for Basic Operations](https://in.mathworks.com/help/matlab/matlab_prog/compatible-array-sizes-for-basic-operations.html).  **Decision Branching**  The body of an if block is only executed if the condition is true.  **For Loops**  When this code is run, the loop body will be executed three times, as the loop counter (c) progresses through the values 1:3 (1, 2, and 3). Basic syntax  | **Example** | **Description** | | --- | --- | | [x = pi](https://www.mathworks.com/help/matlab/matlab_env/create-and-edit-variables.html) | Create variables with the equal sign (=). The left-side (x) is the variable name containing the value on the right-side (pi). | | [y = sin(-5)](https://www.mathworks.com/help/matlab/learn_matlab/calling-functions.html) | You can provide inputs to a function using parentheses. |    Desktop management  | **Function** | **Example** | **Description** | | --- | --- | --- | | [save](https://www.mathworks.com/help/matlab/ref/save.html) | save data.mat | Save your current workspace to a MAT-file. | | [load](https://www.mathworks.com/help/matlab/ref/load.html) | load data.mat | Load the variables in a MAT-file to the Workspace. | | [clear](https://www.mathworks.com/help/matlab/ref/clear.html) | clear | Clear all variables from the Workspace. | | [clc](https://www.mathworks.com/help/matlab/ref/clc.html) | clc | Clear all text from the Command Window. | | [format](https://www.mathworks.com/help/matlab/ref/format.html) | format long | Change how numeric output is displayed. |    Array types  | **Example** | **Description** | | --- | --- | | 4 | scalar | | [3 5] | row vector | | [1;3] | column vector | | [3 4 5;6 7 8] | matrix |    Evenly-spaced vectors  | **Example** | **Description** | | --- | --- | | 1:4 | Create a vector from 1 to 4, spaced by 1, using the [colon (:)](https://www.mathworks.com/help/matlab/ref/colon.html) operator. | | 1:0.5:4 | Create a vector from 1 to 4, spaced by 0.5. | | [linspace](https://www.mathworks.com/help/matlab/ref/linspace.html)(1,10,5) | Create a vector with 5 elements. The values are evenly spaced from 1 to 10. |    Creating matrices  | **Example** | **Description** | | --- | --- | | [rand](https://www.mathworks.com/help/matlab/ref/rand.html)(2) | Create a square matrix with 2 rows and 2 columns. | | [zeros](https://www.mathworks.com/help/matlab/ref/zeros.html)(2,3) | Create a rectangular matrix with 2 rows and 3 columns. |    Indexing  | **Example** | **Description** | | --- | --- | | A([end](https://www.mathworks.com/help/matlab/ref/end.html),2) | Access the element in the second column of the last row. | | A(2,:) | Access the entire second row | | A(1:3,:) | Access all columns of the first three rows. | | A(2) = 11 | Change the value of the second element an array to 11. |    Array operations  | **Example** | **Description** | | --- | --- | | [1 1; 1 1]\*[2 2;2 2]  ans =  4 4  4 4 | Perform [matrix multiplication](https://www.mathworks.com/help/matlab/matlab_prog/array-vs-matrix-operations.html#btyv9yp-4). | | [1 1; 1 1].\*[2 2;2 2]  ans =  2 2  2 2 | Perform [element-wise multiplication](https://www.mathworks.com/help/matlab/matlab_prog/array-vs-matrix-operations.html#bu90xxy-1). |    Multiple outputs  | **Example** | **Description** | | --- | --- | | [xrow,xcol] = [size](https://www.mathworks.com/help/matlab/ref/size.html#bvfgzsm-6)(x) | Save the number of rows and columns in x to two different variables. | | [xMax,idx] = [max](https://www.mathworks.com/help/matlab/ref/max.html)(x) | Calculate the maximum value of x and its corresponding index value. |    Documentation  | **Example** | **Description** | | --- | --- | | [doc](https://www.mathworks.com/help/matlab/ref/doc.html) randi | Open the documentation page for the randi function. |    Plotting  | **Example** | **Description** | | --- | --- | | [plot](https://www.mathworks.com/help/matlab/ref/plot.html)(x,y,"ro-","LineWidth",5) | Plot a red (r) dashed (--) line with a circle (o) marker, with a heavy line width. | | [hold](https://www.mathworks.com/help/matlab/ref/hold.html) on | Add the next line to existing plot. | | hold off | Create a new axes for the next plotted line. | | [title](https://www.mathworks.com/help/matlab/creating_plots/add-title-axis-labels-and-legend-to-graph.html)("My Title") | Add a label to a plot. |    Using tables  | **Example** | **Description** | | --- | --- | | [data.HeightYards](https://www.mathworks.com/help/matlab/matlab_prog/access-data-in-a-table.html) | Extract the variable HeightYards from the table data. | | data.HeightMeters = data.HeightYards\*0.9144 | Derive a table variable from existing data. |    Logicals  | **Example** | **Description** | | --- | --- | | [[5 10 15] > 12](https://www.mathworks.com/help/matlab/matlab_prog/array-comparison-with-relational-operators.html) | Compare a vector to the value 12. | | [v1(v1 > 6)](https://www.mathworks.com/help/matlab/matlab_prog/find-array-elements-that-meet-a-condition.html) | Extract all elements in v1 that are greater than 6. | | x(x==999) = 1 | Replace all values in x that are equal to 999 with the value 1. |    Programming  | **Example** | **Description** | | --- | --- | | [if](https://www.mathworks.com/help/matlab/ref/if.html) x > 0.5  y = 3  else  y = 4  end | If x is greater than 0.5, set the value of y to 3.  Otherwise, set the value of y to 4. | | [for](https://www.mathworks.com/help/matlab/ref/for.html) c = 1:3  disp(c)  end | The loop counter (c) progresses through the values 1:3 (1, 2, and 3).  The loop body displays each value of c. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date:** | **8/July/2020** | **Name:** | **Prashantha naik** | |
| **Course:** | **Introduction to Internet of Things** | **USN:** | **4al17ec074** | |
| **Topic:** | **Chapter 3: Everything Generates Data** | **Semester&Section:** | **6th b** | |
| **Git hub repository** | **prashanth\_couse** |  |  | |
| **AFTERNOON SESSION DETAILS** | | | |
| **Image of session** | | | |
| **Report – Report can be typed or hand written for up to two pages.** **What is Big Data?** Data is information that comes from a variety of sources, such as people, pictures, text, sensors, and web sites. Data also comes from technology devices like cell phones, computers, kiosks, tablets, and cash registers. Most recently, there has been a spike in the volume of data generated by sensors. Sensors are now installed in an ever growing number of locations and objects. These include security cameras, traffic lights, intelligent cars, thermometers, and even grape vines!  Big Data is a lot of data, but what is a lot? No one has an exact number that says when data from an organization is considered “Big Data.” Here are three characteristics that indicate an organization may be dealing with Big Data:   * They have a large amount of data that increasingly requires more storage space (volume). * They have an amount of data that is growing exponentially fast (velocity). * They have data that is generated in different formats (variety).   How much data do sensors collect? Here are some estimated examples:   * Sensors in one autonomous car can generate 4,000 gigabits (Gb) of data per day. * An Airbus A380 Engine generates 1 petabyte (PB) of data on a flight from London to Singapore. * Safety sensors in mining operations can generate up to 2,4 terabits (TB) of data every minute. * Sensors in one smart connected home can produce as much as 1 gigabyte (GB) of information a week.   While Big Data does create challenges for organizations in terms of storage and analytics, it can also provide invaluable information to fine-tune operations and improve customer satisfaction. **Why Do Businesses Analyze Data?** Every organization must become more efficient and innovative to stay competitive and relevant in the digitized world. The IoT is an integral part of achieving that efficiency and innovation.  The goal of many businesses is to collect and analyze the massive amounts of new product-usage data to gain valuable insights. Data analytics allows businesses to better understand the impact of their products and services, adjust their methods and goals, and provide their customers with better products faster. The ability to gain new insights from their data brings value to the business.  To businesses, data is the new oil. Like crude oil, it is valuable, but if it is unrefined it cannot be easily used. Crude oil has to be changed to gasoline, plastic, chemicals, and other substances to create a valuable product. It is the same with data. Data must be broken down and analyzed for it to have value. **Sources of Information** The source of data in the large datasets is varied. Apart from sensor data, other data originates from anything that has been scanned, entered, and released to the Internet from sources such as:   * Social media sites - Facebook, YouTube, eHarmony, and Twitter * HTTP, Web pages, and search engines on the Internet * Historical data from public and private archives * Metadata that is attached to emails, transmitted documents, and pictures * Medical forms, insurance forms, and tax forms * Genomics research using DNA   Collected data can be categorized as structured or unstructured.  Structured data is created by applications that use “fixed” format input such as spreadsheets or medical forms. Even if data is considered structured, different applications create files in different formats that are not necessarily compatible with one another. Structured data may need to be manipulated into a common format such as CSV.  Comma-separated values (CSV) files are a type of plaintext file that use commas to separate columns in a table of data, and the carriage return character to separate rows. Each row is a record. Although they are commonly used for importing and exporting in traditional databases and spreadsheets, there is no specific standard. JSON and XML are also plaintext file types that use a standard way of representing data records. These file formats are compatible with a wide range of applications. Converting data into a common format is a valuable way to combine data from different sources.  Unstructured data is generated in a “freeform” style such as audio, video, web pages, and tweets. Unstructured data requires different tools to prepare data for processing or analysis. The following are two examples:   * Web pages are created to provide data to humans, not machines. “Web scraping” tools automatically extract data from HTML pages. This is similar to a Web Crawler or spider of a search engine. It explores the web to extract data and create the database to respond to the search queries. Web scraping software may use Hypertext Transfer Protocol or a web browser to access the World Wide Web. Typically, web scraping is an automated process which uses a bot or web crawler to do data mining. Specific data is gathered and copied from the web to a database or spreadsheet. The data can then be easily analyzed. * Many large web service providers such as Facebook provide standardized interfaces to collect the data automatically using application programming interfaces (APIs). The most common approach is to use RESTful APIs. RESTful APIs use HTTP as the communication protocol and JSON structure to encode the data. Internet websites like Google and Twitter gather large amounts of static and time series data. Knowledge of the APIs for these sites allow data analysts and engineers to access the large amounts of data that are constantly being generated on the Internet.   Big Data usually has three characteristics. It is a large amount of data that increasingly requires more storage space (volume), that is growing exponentially fast (velocity), and that is generated in different formats (variety).  Fog computing is an architecture that utilizes end-user clients or “edge” devices to do a substantial amount of the pre-processing and storage required by an organization. Fog computing was designed to keep the data closer to the source for pre-processing. | | | |