**DAILY ASSESSMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **07/07/2020** | **Name:** | **Gaganashree P** |
| **Course:** | **Matlab from mathworld** | **USN:** | **4AL15EC 024** |
| **Topic:** | **Indexing and Modifying of Arrays,array calculation,plotting data,obtaining help** | **Semester & Section:** | **8th - A** |
| **GitHub Repository:** | **Gaganashree-P** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
|  |

**REPORT –**

**Indexing and Modifying of arrays**

MATLAB classes support object array indexing by default. Many class designs require no modification to this behavior.

Arrays enable you to reference and assign elements of the array using a subscripted notation. This notation specifies the indices of specific array elements. For example, suppose that you create two arrays of numbers (using [randi](https://in.mathworks.com/help/matlab/ref/randi.html) and concatenation).

Create a 3-by-4 array of integers from 1 through 9:

A = randi(9,3,4)

A =

4 8 5 7

4 2 6 3

7 5 7 7

Create a 1-by-3 array of the numbers 3, 6, 9:

B = [3 6 9];

Reference and assign elements of either array using index values in parentheses:

B(2) = A(3,4);

B

B =

3 7 9

When you execute a statement that involves indexed reference:

C = A(3,4);

MATLAB calls the built-in subsref function to determine how to interpret the statement. Similarly, if you execute a statement that involves indexed assignment:

C(4) = 7;

MATLAB calls the built-in subsasgn function to determine how to interpret the statement.

The MATLAB default subsref and subsasgn functions also work with user-defined objects. For example, create an array of objects of the same class:

for k=1:3

objArray(k) = MyClass;

end

Referencing the second element in the object array, objArray, returns the object constructed when k = 2:

D = objArray(2);

class(D)

ans =

MyClass

You can assign an object to an array of objects of the same class, or an uninitialized variable:

newArray(3,4) = D;

Arrays of objects behave much like numeric arrays in MATLAB. You do not need to implement any special methods to provide standard array behavior with your class.

For general information about array indexing, see [Array Indexing](https://in.mathworks.com/help/matlab/math/array-indexing.html).

### What You Can Modify

You can modify your class indexed reference and/or assignment behavior by implementing class methods called [subsref](https://in.mathworks.com/help/matlab/ref/subsref.html) and [subsasgn](https://in.mathworks.com/help/matlab/ref/subsasgn.html). For syntax description, see their respective reference pages.

Once you add a subsref or subsasgn method to your class, then MATLAB calls only the class method, not the built-in function. Therefore, your class method must implement all the indexed reference and assignment operations that you want your class to support. These operations include:

* Dot notation calls to class methods
* Dot notation reference and assignment involving properties
* Any indexing using parentheses '()'
* Any indexing using braces '{}'

Implementing subsref and subsasgn methods gives you complete control over the interpretation of indexing expressions for objects of your class. Implementing the extent of behaviors that MATLAB provides by default is nontrivial.

### When to Modify Indexing Behavior

Default indexing for object arrays and dot notation for access to properties and methods enables user-defined objects to behave like built-in classes. For example, suppose that you define a class with a property called Data that contains an array of numeric data.

This statement:

obj.Data(2,3)

Returns the value contained in the second row, third column of the array. If you have an array of objects, use an expression like:

objArray(3).Data(2,3)

This statement returns the value contained in the second row, third column of the third element in the array.

Modify the default indexing behavior when your class design requires behavior that is different from MATLAB default behavior.

### Built-In subsref and subsasgn Called in Methods

MATLAB does not call class-defined subsref or subsasgn methods within the overloaded methods. Within class methods, MATLAB always calls the built-in subsref and subsasgn functions. This behavior occurs within the class-defined subsref and subsasgn methods too.

For example, within a class method, this dot reference:

obj.Prop

calls the built-in subsref function. To call the class-defined subsref method, use:

subsref(obj,substruct('.','Prop'))

Whenever a method requires the functionality of the class-defined subsref or subsasgn method, the class must call the overloaded methods as functions. Do not use the operators, '()', '{}', or '.'.

For example, suppose that you define a class to represent polynomial. This class has a subsref method that evaluates the polynomial with the value of the independent variable equal to the subscript. Assume that this statement defines the polynomial with its coefficients:

p = polynom([1 0 -2 -5]);

The MATLAB expression for the resulting polynomial is:

x^3 - 2\*x - 5

This subscripted expression returns the value of the polynomial at x = 3:

p(3)

ans =

16

Suppose that you want to use this feature in another class method. To do so, call the subsref function directly. The evalEqual method accepts two polynom objects and a value at which to evaluate the polynomials:

methods

function ToF = evalEqual(p1,p2,x)

% Create arguments for subsref

subs.type = '()';

subs.subs = {x};

% Need to call subsref explicitly

y1 = subsref(p1,subs);

y2 = subsref(p2,subs);

if y1 == y2

ToF = true;

else

ToF = false;

end

end

end

This behavior enables you to use standard MATLAB indexing to implement specialized behaviors. See [Class with Modified Indexing](https://in.mathworks.com/help/matlab/matlab_oop/class-with-modified-indexing.html) for examples of how to use both built-in and class-modified indexing.

### Avoid Overriding Access Attributes

Because subsref is a class method, it has access to private class members. Avoid inadvertently giving access to private methods and properties as you handle various types of reference. Consider this subsref method defined for a class having private properties, x and y:

classdef MyPlot

properties (Access = private)

x

y

end

properties

Maximum

Minimum

Average

end

methods

function obj = MyPlot(x,y)

obj.x = x;

obj.y = y;

obj.Maximum = max(y);

obj.Minimum = min(y);

obj.Average = mean(y);

end

function B = subsref(A,S)

switch S(1).type

case '.'

switch S(1).subs

case 'plot'

% Reference to A.x and A.y call built-in subsref

B = plot(A.x,A.y);

otherwise

% Enable dot notation for all properties and methods

B = A.(S.subs);

end

end

end

end

end

This subsref enables the use of dot notation to create a plot using the name 'plot'. The statement:

obj = MyPlot(1:10,1:10);

h = obj.plot;

calls the [plot](https://in.mathworks.com/help/matlab/ref/plot.html) function and returns the handle to the graphics object.

You do not need to code each method and property name. The otherwise code in the inner switch block manages any name reference that you do not explicitly specify in case statements. However, using this technique exposes any private and protected class members via dot notation. For example, you can reference the private property, x, with this statement:

obj.x

ans =

1 2 3 4 5 6 7 8 9 10

**DAILY ASSESSMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **7/07/2020** | **Name:** | **Gaganashree P** |
| **Course:** | **IOT** | **USN:** | **4AL15EC024** |
| **Topic:** | **Chapter 3 and 4** | **Semester & Section:** | **8TH SEM &A Section** |
| **Github Repository:** | **Gaganashree-P** |  |  |

|  |
| --- |
| **AFTERNOON SESSION DETAILS**   ****IoT:Everything can be automated**** The term, Internet of Things (IoT) has been lingering over almost every industry for many years now. This topic of conversation has shifted from what once was a buzzword into what is now, without a doubt, a reality. A plethora of new and exciting IoT devices and business applications are emerging at an exponential rate each day. Although this may be the case, very few businesses understand how they can benefit from the Internet of Things.  The lack of enthusiasm to delve into the connected world may be caused by a number of factors. Some business owners have concerns around the security of IoT, and some are simply set in their ways. But we found that the most common reason for not giving IoT a second look is the lack of education. If you do a quick search for “examples of IoT” you’ll find the same example across the board, the connected coffee scenario.  The connected coffee example simply explains how your morning alarm or mobile phone can trigger your coffee machine. Although it’s a satisfying gimmick to wake up to a steaming cup of coffee, it won’t cause any business owner to invest their time and money into this growing technology. Admittedly we’re guilty of using this example too, but decision makers deserve to understand the business benefits of utilising IoT. That’s why we’ve taken the time to collate some of the main advantages for companies around the world.  IoT and Automation  The Internet of Things (IoT) is a network of connected… well, ‘things’. These ‘things’ (devices) communicate with each other using [machine to machine communication (M2M)](https://www.infisim.com/m2m-sim-cards/). Information is traversed between devices so that processes can be automated, without the need for human intervention. By reducing the number of people involved in a business process, several advantages arise, including improved accuracy and up-time.  IoT and Accuracy  For many businesses minimising errors is a main priority, and whether we like it or not people make mistakes. [Human error](https://www.smartindustry.com/blog/smart-industry-connect/can-the-iot-eliminate-human-error/) is something that is often overlooked or discounted when refining processes. By connecting machines and allowing them to control themselves automatically, fewer members of staff are needed to complete certain tasks.  Increase up-time with the IoT  Another advantage of automating your business processes using IoT is an increase in up-time. But what do we mean by this?  In this instance, up-time is the amount of time your business is open or running for. As we all know, your staff don’t work around the clock, they come into work at a specific time and leave at another. This means that your business is only open for a certain number of hours per day (unless your staff’s shifts overlap). Machines on the other hand aren’t restricted by time. Automating your business processes via the Internet of Things will allow them to run 24/7.  Although there are a lot of ethical arguments that arise when it comes to automation of business (due to the ‘loss’ of jobs), there needn’t be. We aren’t suggesting you make your entire workforce redundant and replace them with robot workers. Human jobs will definitely remain a necessity as machines need to be programmed and maintained.  In traditional businesses engineers are often employed to fix faulty machinery and devices if something goes wrong. Unfortunately, this usually involves a lot of testing to find the faulty component. A lot of time is wasted during this process and your business is paying for that loss of efficiency.  When a process is automated your own network of IoT devices will notify your engineers of exactly where and what the problem is. This way the problem can be identified and fixed quickly, reducing the amount of down time you would have previously experienced.  IoT Analysis and Monitoring  The beauty of the Internet of Things (IoT) isn’t necessarily the improved performance of business processes. In a modern-day business, analytics and logistics are becoming more and more important as technology develops. Understanding why a problem has occurred or identifying areas of improvement is getting much easier because of this.  A great example of how [IoT can be used to monitor business processes](https://www.infisim.com/nova-m2m-sim-management/) is within the fleet management industry. By equipping each vehicle within a fleet with a GPS Tracker and an M2M SIM card, each vehicle can be located and monitored in real-time.  Almost any process can be monitored with the help of IoT as long as you have the right sensors and equipment. The same goes for remotely accessing devices such as CCTV and digital signage. |