CN208
Introductory Computer Programming

Week 5:- Introduction to MATLAB

By

Dr. Piya Techateerawat

Introduction to MATLAB

- **Programming Language Background**
- Basic Data Manipulation
- The MATLAB User Interface
- Scripts

Programming Language Background

- Abstraction
  - Data
  - Procedure
- Algorithms
- Programming Paradigms
  - Functional programming
  - Procedural programming
  - Object-oriented programming (OOP)

Introduction to MATLAB

- Programming Language Background
- **Basic Data Manipulation**
- The MATLAB User Interface
- Scripts

Basic Data Manipulation

- Starting and Stopping MATLAB
- Assigning Values to Variables
  - z = x+ y
  - z = 4*x – y
- Data Typing
  - temp = 35
  - temp + 2

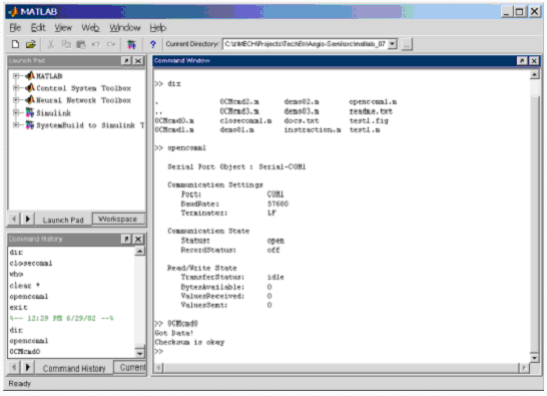
    ans =

    37
  - temp = ‘ unit is C’
  - temp + 1

Introduction to MATLAB

- Programming Language Background
- Basic Data Manipulation
- **The MATLAB User Interface**
- Scripts

The MATLAB User Interface



Introduction to MATLAB

- Programming Language Background
- Basic Data Manipulation
- The MATLAB User Interface
- **Scripts**

Scripts

- Creating text files “m-files”
- Creating scripts
  - Clear
  - Clc
  - A= 3;
  - ….
- Current Directory
- Running Script
  - Debugging Script

Scripts

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%(6) Creating scripts or functions using m-files:
%
% Matlab scripts are files with ".m" extension containing Matlab
% commands.  Variables in a script file are global and will change the
% value of variables of the same name in the environment of the current
% Matlab session.  A script with name "script1.m" can be invoked by
% typing "script1" in the command window.

% Functions are also m-files. The first line in a function file must be
% of this form:
% function [outarg_1, ..., outarg_m] = myfunction(inarg_1, ..., inarg_n)
%
% The function name should be the same as that of the file
% (i.e. function "myfunction" should be saved in file "myfunction.m").
% Have a look at myfunction.m and myotherfunction.m for examples.
%
% Functions are executed using local workspaces: there is no risk of
% conflicts with the variables in the main workspace. At the end of a
% function execution only the output arguments will be visible in the
% main workspace.

a = [1 2 3 4];              % Global variable a
b = myfunction(2 * a)       % Call myfunction which has local
                            %   variable a
a                           % Global variable a is unchanged

[c, d] = ...
  myotherfunction(a, b)     % Call myotherfunction with two return
                            % values
```

Reference

- http://www.cs.brown.edu/courses/cs143/MatlabTutorialCode.html

Q & A