



SMART TRAINERS

LESSON PLAN FOR MATLAB PROGRAMMING COURSE

COURSE TITLE:	Basic to Intermediate MATLAB Programming		
COURSE CODE:	MAT 001		
CREDIT HOURS:	02		
DURATION:	08 Weeks (03 Classes Each Week)		
COURSE INSTRUCTOR:	Batch:	Course Starting Date:	Course Suspension Date:

COURSE LEARNING OUTCOMES: Upon successful completion of the course, the student will be able to: Navigate the MATLAB desktop environment and write, run, and debug scripts and functions using the MATLAB IDE and Command Window. Work confidently with MATLAB's core data types — scalars, vectors, matrices, strings, cell arrays, and structures — and apply built-in mathematical and statistical functions. Implement control flow (conditionals, loops) and write reusable functions, anonymous functions, and recursive algorithms. Produce publication-quality 2D and 3D plots, graphs, and visualisations using MATLAB's plotting and graphics tools. Perform numerical methods — linear algebra, interpolation, differentiation, integration, and ODE solving — using MATLAB's toolboxes. Read, write, and process data from files (CSV, Excel, text) and apply signal processing and image processing fundamentals. Build interactive GUIs using App Designer and apply Simulink for basic model-based design and simulation.

LESSON CONTENTS AND ASSOCIATED CLO(s)		
CLO	Description	Level
01	Navigate the MATLAB IDE; create and manipulate scalars, vectors, matrices, strings, cell arrays, and structures using built-in operators and functions.	C1
02	Implement control flow (if/elseif/else, for, while, switch) and write user-defined functions, anonymous functions, and recursive algorithms.	C2
03	Create 2D/3D plots, customise figures (labels, legends, subplots, colour maps), and export publication-quality graphics.	C2
04	Apply numerical methods: linear algebra (solve, eig, svd), interpolation, numerical differentiation/integration, and ODE solvers (ode45, ode23).	C3
05	Read and write data from CSV, Excel, and text files; apply signal processing fundamentals (FFT, filtering) and basic image processing operations.	C3
06	Build interactive GUIs with MATLAB App Designer and construct, simulate, and analyse basic Simulink block-diagram models.	C4

Contents	Week	CLO No.	Marks Assigned	Delivery Methods	Assessment Methods
PHASE 1 — BASIC MATLAB (Weeks 1 – 4)					
<p>1. MATLAB Environment & Fundamentals</p> <ul style="list-style-type: none"> • MATLAB desktop: Command Window, Workspace, Editor, File Browser • Running commands vs. scripts vs. live scripts (.mlx) • Variables, naming conventions, and the semicolon suppression rule • Built-in constants: pi, eps, inf, NaN, true, false • Getting help: doc, help, lookfor, and MathWorks documentation <p>2. Scalars, Vectors & Matrices</p> <ul style="list-style-type: none"> • Creating row and column vectors; linspace and logspace • Matrix construction: zeros, ones, eye, rand, randn, magic • Indexing and slicing — single, range, end, logical indexing • Element-wise vs. matrix arithmetic (.*, ./, .^) • Transpose, determinant, inverse, rank, trace 	Week 1	01	5%	Online Lectures & Practical Work	Practical Project
<p>3. Strings, Cell Arrays & Structures</p> <ul style="list-style-type: none"> • Character arrays vs. string objects (double-quoted strings) • String functions: length, numel, strcmp, strcat, strsplit, sprintf • Cell arrays: creating, indexing with {} and () • Struct arrays: fieldnames, rmfield, orderfields • Converting between data types: num2str, str2num, cell2mat <p>4. Control Flow — Conditionals & Loops</p> <ul style="list-style-type: none"> • if / elseif / else / end and nested conditionals • switch / case / otherwise 	Week 2	01–02	5%	Online Lectures & Practical Work	Practical Project

<ul style="list-style-type: none"> • for loop: iterating over ranges, vectors, and cell arrays • while loop and do-while pattern with break / continue • Vectorisation vs. loops — performance considerations 					
<p>5. Functions — User-Defined & Anonymous</p> <ul style="list-style-type: none"> • Defining functions in separate .m files (function...end) • Input/output arguments: nargin, nargout, varargin, varargout • Local functions, nested functions, and scope • Anonymous functions: @(x) syntax and closures • Recursive functions with base-case design <p>6. Built-in Math & Statistical Functions</p> <ul style="list-style-type: none"> • Trigonometric, exponential, logarithmic functions • Rounding: floor, ceil, round, fix, mod, rem • Statistical: mean, median, std, var, min, max, sum, prod • Sorting and searching: sort, sortrows, find, unique, ismember • Random number generation and setting seed (rng) 	Week 3	02	10%	Online Lectures & Practical Work	Practical Project
<p>7. 2D Plotting & Visualisation</p> <ul style="list-style-type: none"> • plot, stem, bar, histogram, scatter, pie, area • Customising: title, xlabel, ylabel, legend, grid, axis • Line styles, marker types, and colour specifications • Multiple plots: hold on/off and subplot(m,n,p) • Saving figures: saveas, exportgraphics (PDF, PNG, SVG) <p>8. 3D Plotting & Advanced Graphics</p> <ul style="list-style-type: none"> • plot3, surf, mesh, contour, contourf • Colour maps: colormap, colorbar, shading interp • Viewing angles: view, rotate3d, camlight • Quiver plots and vector field visualisation • Animating plots with drawnow and gif export 	Week 4	02–03	10%	Online Lectures & Practical Work	Practical Project
PHASE 2 — INTERMEDIATE MATLAB (Weeks 5 – 8)					

<p>9. Linear Algebra & Matrix Operations</p> <ul style="list-style-type: none"> • Solving linear systems: $A \setminus b$, $\text{inv}(A)$, $\text{pinv}(A)$ • Eigenvalues and eigenvectors: eig, eigs • Singular Value Decomposition: svd and applications • LU, QR, and Cholesky decomposition • Norms, condition number, and numerical stability <p>10. Interpolation, Differentiation & Integration</p> <ul style="list-style-type: none"> • Polynomial fitting: polyfit, polyval • Interpolation: interp1 (linear, spline, pchip), interp2 • Numerical differentiation: diff, gradient • Numerical integration: trapz, cumtrapz, integral, integral2 • Root finding: fzero, fsolve, roots 	Week 5	04	10%	Online Lectures & Practical Work	Practical Project
<p>11. ODE Solving & Optimisation</p> <ul style="list-style-type: none"> • Solving ODEs with ode45, ode23, ode15s (stiff systems) • Setting ODE options with odeset (tolerances, events) • Plotting ODE solutions and phase portraits • Unconstrained optimisation: fminbnd, fminsearch • Linear programming with linprog (Optimisation Toolbox) <p>12. File I/O & Data Import / Export</p> <ul style="list-style-type: none"> • Reading CSV and delimited text: readmatrix, readtable, textscan • Writing data: writematrix, writetable, fprintf • Excel import/export: readtable with .xlsx, xlsread/xlswrite • Loading and saving MATLAB workspaces: load, save (.mat) • Web data access: webread, jsondecode (REST APIs) 	Week 6	04–05	10%	Online Lectures & Practical Work	Practical Project

<p>13. Signal Processing & Image Processing</p> <ul style="list-style-type: none"> • Discrete Fourier Transform: fft, ifft, fftshift, fftfreq • Power spectrum, frequency analysis, and spectrogram • Digital filtering: FIR/IIR design with butter, fir1, filter • Image I/O: imread, imwrite, imshow, imresize, imcrop • Image operations: rgb2gray, edge, imadjust, imrotate, imfilter <p>14. App Designer & GUI Development</p> <ul style="list-style-type: none"> • Launching App Designer and exploring the canvas/component library • Adding UI components: buttons, sliders, axes, text fields, dropdowns • Callbacks: writing and connecting event-driven code • Sharing data between callbacks using app properties • Packaging and deploying standalone apps (.mlapp, compiler) 	Week 7	05–06	20%	Online Lectures & Practical Work	Practical Project
<p>15. Simulink — Model-Based Design</p> <ul style="list-style-type: none"> • Simulink environment: Library Browser, block diagram basics • Continuous and discrete simulation: Source, Sink, Math blocks • Building a PID control system model from scratch • Scope and To-Workspace blocks for result extraction • Running simulations and tuning parameters interactively <p>16. Capstone Projects & Course Review</p> <ul style="list-style-type: none"> • Capstone A: Numerical solver — beam deflection / heat equation • Capstone B: Signal analysis dashboard with interactive App Designer GUI • Capstone C: Simulink control system — DC motor speed control • Code review, best practices, and debugging strategies • Pathways: Simulink Control, Deep Learning Toolbox, Python-MATLAB integration 	Week 8	05–06	30%	Online Lectures & Practical Work	Practical Project