

**ES106: PROGRAMMING FOR ENGINEERS  
CREDITS - 3 (LTP:1,0,2)**

**Course Objectives:**

To enhance logical thinking and to impart basic programming skills.

**Teaching and Assessment Scheme:**

Teaching Scheme (Hours per week)			Credits	Assessment Scheme				Total Marks
L	T	P		Theory		Practical		
			ESE	CE	ESE	CE		
1	0	4	3	30	20	40	60	150

**Course Contents:**

Unit No.	Topics	Teaching Hours
1	<b>Introduction to Programming</b> Introduction to the idea of algorithm; Introduction to Programming (Flow chart/pseudocode); Computing Software: System Software, Languages, Tools;	2
2	<b>Vectors and Matrices</b> Creations of Vectors and Matrices, Mathematical Operations with Vectors and Matrices: Addition, Multiplication, Determinants, Matrix Inverse; Data Input/Output: Entering a Scalar, String, Vector and Matrix; Input/Output Data files	3
3	<b>Program Flow Control</b> Logical Operators, Writing and evaluation of conditionals and consequent branching, Iteration and loops.	4
4	<b>Basic Algorithm</b> Searching, Basic Sorting Algorithms (Bubble, Insertion and Selection), Finding roots of equations, notion of order of complexity through example programs	5
5	<b>Functions</b> Functions (including using built in libraries), Function File, Sub function, Anonymous Function, Inline Function, Passing Array to function	Laboratory
6	<b>Plotting and Graphics</b> 2D Plotting: Annotations and Enhancements, Interactive Plotting, Animation; 3D Plotting: Lines, Surfaces;	Laboratory
<b>Total</b>		<b>14</b>

**List of References:**

- Edward B. Magrab and at. al., “*An Engineer’s Guide to Matlab*”, Prentice Hall
- Brian D. Hahn and Daniel T. Valentine, “*Essential MATLAB for Engineers and Scientists*”, Third Edition, ELSEVIER
- E. Balaguruswamy, “*Programming in ANSI C*”, Tata McGraw-Hill

**Course Outcomes (COs):**

At the end of this course students will be able to ...

1. Formulate simple algorithms for arithmetic and logical problems.
2. Translate the algorithms to MATLAB programs
3. Use vectors and matrices for mathematical operations
4. Implement conditional branching, iteration and recursions
5. Decompose a problem into functions
6. Apply programming to solve matrix addition and multiplication problems and searching and sorting problems.
7. Apply programming to solve simple numerical method problems, namely root finding of function, differentiation of function and simple integration.