

Module: 1/3

Module name: Linux Systems

Overview	This module is a kick-start course intended to get new programmers up and running with Linux OS. This course will start with basics of Linux and its features, then covers substantial commands that will be used by a user and programmer. You will be learning Shell scripting, followed by programming tools including make, gdb, VIM. This module then deals with compilation stages, coding guidelines and creating projects using makefiles.
Description	If you are not familiar with Linux as a user or a developer, this course is a prerequisite for other courses delivered at EMERTXE. You will get familiar with the command line interface, the shell, and the essential commands which you will use every now and then as a Linux user/programmer. You will get to know the power of Linux by learning redirection, pipes and how to combine multiple commands together to automate tasks. Following this, we teach you about various stages involved in compilation using GCC, and writing Makefiles. Finally you will learn how to modularize and organize "C" source and header files to create projects and libraries .
Objective	<ul style="list-style-type: none"> ● To get familiar with Linux Operating systems, and various commands, and VIM editor. ● You should be able to write shell scripts ● To understand the different stages involved in code compilation. ● You should be able to know how to create projects and automate the build using makefiles.
Platform	Linux
Delivery method	Instructor led, Assignments, Mini Project.
Course topics	Overview of Linux OS, Command Line Interface - Shell, Environment variables, shell commands, filtering commands, user accounts, remote login, redirection, pipes, Directory and File system structure, Visual editor (VIM), Shell scripting, Makefiles, code compilation stages, coding guidelines and creation of project and libraries.

Module: 2/3

Module name: Advanced C

Overview	An assignment filled intense Advanced C programming course, taken from Problem to Logic till Program, concluded by an apt project.
Description	<p>Lot of book tells about what is programming. Many also tell how to write a program. But very few tells you about how to translate logic into a program. Specifically, in this fast paced industry, when you don't have time to think to program, this course comes really handy. It builds on the basics of programming, smooth sailing through the advanced nitty-grittys of the language and how to use it to translate logic, the industry-way.</p> <p>Every class is backed by discussion and topic related assignments. Finally a project letting you apply most of the concepts learned throughout the course.</p>
Objective	<ul style="list-style-type: none"> ● To have a clear concept of C language. ● To obtain good quality and style in programming. ● To induce confidence in candidates.
Platform	Any (with specific mention to Linux)
Delivery method	Instructor led, Assignments, Mini Project.
Course topics	Basics, Operators, Conditionals, Arrays, Pointers, Structures, Unions, Bit-wise operations, Functions, Files, Preprocessor directives, Recursion, Creating & Building a project, Makefiles, Logic to program translation, Creating your own library, Introduction to Data Structures.

Module: 3/3

Module name: Micro controllers

Overview	A practical approach on Micro controller with basic Hardware concept and Embedded C programming. A Practical approach on Micro-controller based application development with Basic Hardware concepts, Peripheral Programming, Protocols concluded by Project.
Description	<p>A complete course module dealing with how to write a embedded C program for a Hardware with any micro controller on it. Planning on the software and hardware optimization in application design for ease in development.</p> <p>This module is dealt with practical issues normally faced in development phase and how to overcome them. Building an complete application with the help of gained knowledge throughout the course as a project.</p>
Objective	<ul style="list-style-type: none"> ● To provide a basic idea of hardware/electronics aspects of programming, which an embedded engineer requires. ● On completion of this module the candidate should be able to program any micro controller and design basic application with ease.
Platform	<p>Software / Tools: Any (with specific mention to Linux)</p> <p>Hardware: 8051 and PIC based Architecture</p>
Delivery method	Instructor led, Assignments, Mini Project.
Course topics	Introduction to embedded systems, Microprocessors vs Micro controllers, Each session with hardware related concept on which the candidate will be working on, GPIOs, Analog I/Os, Types of memories and its usage ,interfacing etc., Basic micro controller peripherals such as Timers, Counters etc, Interrupts and its sources, Basic communication protocols like UART, SPI, I2C etc.