

## Course: Computing Systems

### Syllabus

Week	Module	Description
1	Getting Started with Programming	This module introduces learners to the fundamental concepts of computing Systems. It covers the components of computers - both hardware and software. It also discusses different architectural models like Von-Neumann, Harvard and Modified Harvard architectures.
2	Computer System Architecture - Memory and I/O	This module explores how Memory and I/O help in improving Architecture performance.
3	Introduction to an Operating System	This module builds a strong foundation in Operating System basics, commands and command line interfaces. The module also covers basic UNIX commands like ls, pwd, cd, mkdir, passwd, man, info, date and cal. The module also covers text processing commands like cat, cp, mv, rm, chmod, IO redirection, filters - wc, sort, head, tail, grep, find, piping, hard link, and symbolic link. The module also covers how to write Shell scripts with shell environments and variables.
4	Shell Scripting	This module covers all constructs like conditional, looping etc. in Shell scripting. It also covers file handling in Shell scripting. The module solves multiple problems using Shell scripting.
5	Program Execution in a System	This module covers the difference between Program and Process. It also covers light weight processes (threads) and its merits compared to conventional processes. The module also discusses the mapping between user level and kernel level. Data sharing across multiple processes / threads will also be discussed. It also covers the performance optimization techniques like thread pools and zeroed out threads. Basic issues related to data concurrency and sharing (synchronization) will also be discussed.
6	Virtual Environment and Isolation	This module introduces the concept of virtualization, illustrates the concepts with examples at different levels and highlights commonly used models of virtualization. The module starts by differentiating between abstraction and virtualization, introduces notions of physical vs virtual resources as well as

		physical vs interfaces and how these manifest in the context of computer systems. Classification of virtual environments as process virtual machines and system virtual machines is explained along with their primary function of isolation. High-level Language Virtual Machines as a special case of Process Virtual Machines, and Python Virtual Environment as an example of the same are explained further. Containers as light-weight System Virtual Machines are introduced.
7	Source Code Repositories and Versioning Systems	This module motivates the need for a common repository of code to be used by a team of developers as well as the need for developing and tracking versions of the same software. The module briefly describes and explains the features of typical (software) tools/technologies used as code repository systems and for versioning systems
8	Profiling and Debugging	This module introduces the concepts of profiling a program at run-time and debugging it using typical dynamic profilers and debuggers.
9	Introduction to GPUs	This module introduces GPUs as co-processors, and the GPGPU architecture.
10	Program Execution and Performance Tuning in GPUs	This module explores ways to tune the performance of multi-threaded programs on GPUs so that they are scalable with the number of processors (i.e. GPU cores)
11	Introduction to Distributed Systems	This module introduces elementary concepts related to systems for distributing data and computation, different ways to structure / architect distributed systems, and data distribution approaches in different scenarios / contexts
12	Introduction to Cloud Computing	This module introduces the notion of centralized computing services in the form of Cloud, justifies the economic / business necessity, briefly explains the underlying technology of virtualization and its implication to elasticity of resource management,
13	Introduction to Mobile/Embedded Computing	This module introduces the notion of computing and computing systems that are mobile and/or embedded physically in no other (possibly, non-computing) device or system. The introduction is at a high level without many technical details with the intent to focus on