NIRMA UNIVERSITY

Institute:	Institute of Technology, School of Technology
Name of Programme:	MTech CSE, MTech CSE (Data Science)
Course Code:	6CS378ME25
Course Title:	High Performance Computing and Architecture
Course Type:	Department Elective-III
Year of Introduction:	2025-26

L	T	Practical Component				C
		LPW	PW	W	S	
3	0	2	-	-	-	3

Course Learning Outcomes (CLO):

At the end of the course, the students will be able to:

1. explain various high-performance computing system architectures (BL2)
2. identify the issues involved in the design of HPC systems (BL3)
3. analyse algorithms and data structures for applications that are efficient on large-scale systems (BL4)
4. design compute-intensive applications on various HPC platforms. (BL6)

Unit	Contents	Teaching Hours (Total 45)
Unit-I	**Parallel Programming Models and Techniques:** Introduction to parallel programming: threads and processes, Models of parallelism: data parallelism, task parallelism, Thread management and synchronization, Basics of OpenMP, Pthreads, and MPI, Shared memory vs distributed memory systems	05
Unit-II	**Requirement and general issues of High Performance Computing:** Introduction to Cluster and Grid computing, Dependable Clustered Computing, Metacomputing: Harnessing Informal Supercomputers, Specifying Resources and Services in Metacomputing Systems, High Speed Networks, Lightweight Messaging Systems, Xpress Transport Protocol, Software RAID and Parallel File systems, Distributed Shared Memory, Load Balancing Over Networks, Job and Resource Management Systems	18
Unit-III	**Multicore Architecture and Programming:** Overview of processor architectures: Single-core vs. multicore, Evolution of multicore processors: Motivation, challenges, and benefits, Multicore processor design principles. Types of multicore architectures: Symmetric Multiprocessing (SMP), Asymmetric Multiprocessing (AMP), NUMA, etc., Multicore processors in modern computing environments.	12
Unit-IV	**CUDA programming:** GPU computing and CUDA programming Multithreading in modern processors (e.g., Intel Hyper-Threading, AMD's Simultaneous Multithreading), Vectorization and SIMD instructions, Fault tolerance in multicore systems	06

Unit-V	**Advance Computing:** Introduction to Petascale computing, Optical Computing, Quantum computing, and its issues.	04

Self-Study:

The self-study contents will be declared at the commencement of the semester. Around 10% of the questions will be asked from self-study content.

Suggested Readings/ References:

1. Gerassimos Barlas, Multicore and GPU Programming: An Integrated Approach, Morgan Kauffman
2. Kai Hwang, N. D. Jotwani, Advance Computer Architecture: Parallelism, Scalability, Programmability, McGraw Hill
3. Buyya, Rajkumar, High Performance Cluster Computing: Programming and Applications, Pearson Education
4. John L. Hennessy, David A. Patterson Computer Architecture: A Quantitative Approach, Morgan Kauffman
5. Quinn, Parallel Programming in C with MPI and OpenMP, TMH
6. Maurice Herlihy, Nir Shavit, The Art of Multiprocessor Programming, Morgan Kauffman
7. Peter Pacheco, Introduction to Parallel Programming, Morgan Kauffman
8. Jason Sanders, Edward Kandrot, CUDA by Example: An Introduction to General-Purpose GPU Programming, NVIDIA.

Suggested List of Experiments:

Sr. No.	Name of Experiments/Exercises	Hours
1	Develop compute-intensive applications using structural programming language and measure execution time along with other performance parameters	02
2	To understand the functioning of Processes, Threads, and speedup, rewrite the same program (as developed in Experiment No 1) by creating child processes. Assign the tasks to threads. Apply concepts of parallel processing and observe performance parameters. Calculate Speedup	04
3	Meta computing: For harnessing computational Power 1. Configure metacomputing middleware (e.g., Alchemi) and verify the execution of built-in applications in terms of their functioning, resource utilization, libraries used, etc. 2. Write a small piece of code that can exploit CPU cycles using Alchemi/Globus libraries and execute in a Meta computing environment. Observe CPU utilization and performance of processes and threads	06
6	Explore and understand the Cluster platform established in CoE. Design and develop cluster compatible Algorithms for assigned tasks (It will be given in a group of 4) and deploy them on the Cluster. Observe the performance and results	06
7	Solve practical engineering applications using CUDA and analyse the performance	06
8	Study any one DSM system in detail. Prepare a report and deliver a presentation (Self Study).	06