algorithms to achieve high performance computing

S. NO	Course Outcomes (CO)
CO1	Understand basic concepts, advanced computer architectures, parallel algorithms
CO2	Apply priciples of Memory Hierarchies, Multi core Processors
CO3	Parallel Programming involving Revealing concurrency
CO4	Compare high performance methods with Achieving Measuring performance

S. NO	Contents	Contact Hours
UNIT 1	Introduction to advanced computer architectures, parallel algorithms, parallel languages, and performance oriented computing, discussing about the key characteristics of highend computing architectures.	**8**

UNIT 2	Introduction to Computational Science and Engineering Applications, their characteristics and requirements,Review of Computational Complexity, Performance: metrics and measurements, Granularity and Partitioning, emporal/spatial/stream/kernel, Basic methods for parallel programming, Real-world case studies which are drawn from multiscale, multidiscipline applications.	**8**
UNIT 3	Memory Hierarchies, Multi core Processors, Homogeneous and Heterogeneous, Sharedmemory Symmetric Multiprocessors, Vector Computers, Distributed Memory Computers, Supercomputers and Petascale Systems, Application Accelerators/ Reconfigurable Computing, Novel computers: Stream, multithreaded, and purposebuilt	**9**
UNIT 4	Parallel Programming involving Revealing concurrency in applications, Task and Functional parallelism, Task Scheduling, Synchronization Methods, Parallel Primitives (collective operations), SPMD Programming (threads, OpenMP, MPI), I/O and File Systems, Parallel Matlabs (Parallel Matlab, StarP, Matlab MPI) Partitioning Global Address Space (PGAS) languages (UPC, Titanium, Global Arrays).	**9**
UNIT 5	Discussion about high performance methods with Achieving Measuring performance, Identifying performance bottlenecks, Restructuring applications for deep memory hierarchies, Partitioning applications for heterogeneous resources, Using existing libraries, tools, and framework.	**8**
	TOTAL	**42**