models like CUDA and OpenCL. Students will learn to develop GPU-accelerated applications, apply GPU computing to solve real-world problems, and evaluate the suitability of GPU computing for various computational tasks.

S. NO	Course Outcomes (CO)
CO1	Understand the fundamental concepts and architecture of GPUs.
CO2	Utilize GPU programming models (e.g., CUDA, OpenCL) to develop efficient GPU-accelerated applications.
CO3	Apply GPU computing techniques to solve real-world problems in various domains.
CO4	Evaluate the suitability of GPU computing for specific computational tasks.

S. NO	Contents	Contact Hours

UNIT 1	**Introduction to GPU Computing:** GPU Architecture: Comparison with CPUs, CUDA Cores, SMs, and Memory Hierarchy, GPU Programming Models: CUDA and OpenCL, Basic Concepts and Syntax Parallel Programming Paradigms: Data Parallelism, Task Parallelism , Hybrid Parallelism	**8**
UNIT 2	**CUDA Programming**: CUDA Programming Fundamentals: Kernels and Device Functions, Memory Management (Host and Device Memory), Thread and Block Hierarchies, CUDA Performance Optimization:, Memory Coalescing, Shared Memory Optimization, Texture, Memory Utilization, Occupancy and Warp Divergence, CUDA Examples:, Matrix Multiplication, Image Processing, Monte Carlo Simulations	**9**
UNIT 3	**OpenCL Programming:** OpenCL Basics: OpenCL Runtime and Host API, OpenCL Kernels and Work Items, Memory Management (Host and Device Memory), OpenCL Performance Optimization:, Work Group and Work Item Scheduling, Data Transfer Optimization, Kernel, Optimization Techniques, OpenCL Examples:, Image Filtering, Molecular Dynamics Simulations, Scientific Computing Applications	**9**
UNIT 4	**Advanced GPU Topics:** GPU Heterogeneous Computing: CPU-GPU Integration, GPU-GPU Communication, GPU Libraries and Frameworks:, CUDA Toolkit, OpenCL Runtime, cuBLAS, cuFFT, cuDNN, GPU Programming Patterns:, Reduction Operations, Scan Operations, Sorting Algorithms, GPU Debugging and Profiling Tools	**8**
UNIT 5	**GPU Applications and Case Studies:** GPU Applications in Various Domains:, Machine Learning and Deep Learning, Computational Fluid Dynamics, Bioinformatics, Financial Modeling, Scientific Visualization, GPU Case Studies:, Real-world examples of GPU-accelerated applications, Performance analysis and benchmarking, Future Trends in GPU Computing	**8**
	TOTAL	**42**

REFERENCES

S.No.	Name of Books/Authors/Publishers	Year of Publication / Reprint