Course code: Course Title	Course Structure			Pre-Requisite
SE429: GPU Computing	L	T	P	NIL
	3	0	2	

Course Objective: To learn parallel programming with Graphics Processing Units (GPUs).

S. NO	Course Outcomes (CO)
CO1	Demonstrate deep understanding of basic concepts of Graphics Processing Units (GPUs), parallel programming models like CUDA and OpenCL.
CO2	Utilize various memory types (global, shared, constant) and synchronization mechanisms to optimize memory allocation.
CO3	Demonstrate the use of device and host functions for efficient GPU programming.
CO4	Identify and resolve parallel programming challenges such as error handling, synchronization issues, and algorithmic efficiency in GPU computing.
CO5	Develop optimized GPU-based solutions for real-world applications.

S. NO	Contents	Contact Hours
UNIT 1	**Introduction:** History, Graphics Processors, Graphics Processing Units, GPGPUs. Clock speeds, CPU / GPU comparisons, Heterogeneity, Accelerators, Parallel programming, CUDA OpenCL / OpenACC, Hello World Computation Kernels, Launch parameters, Thread hierarchy, Warps/ Wavefronts, Thread blocks / Workgroups, Streaming multiprocessors, 1D / 2D / 3D thread mapping, Device properties, Simple Programs.	10
UNIT 2	**Memory:** Memory hierarchy, DRAM / global, local / shared, private / local, textures, Constant Memory, Pointers, Parameter Passing, Arrays and dynamic Memory, Multi-dimensional Arrays, Memory Allocation, Memory copying across devices, Programs with matrices, Performance evaluation with different memories.	8
UNIT 3	**Synchronization:** Memory Consistency, Barriers (local versus global), Atomics, Memory fence. Prefix sum, Reduction. Programs for concurrent Data Structures such as Worklists, Linked-lists. Synchronization across CPU and GPU. Functions: Device functions, Host functions, Kernels functions, Using libraries (such as Thrust), and developing libraries.	9
UNIT 4	**Support:** Debugging GPU Programs. Profiling, Profile tools, Performance aspects. Streams: Asynchronous processing, tasks, Task-dependence, Overlapped data transfers, Default Stream, Synchronization with streams. Events, Event-based-Synchronization - Overlapping data transfer and kernel execution, pitfalls.	8
UNIT 5	**Advanced topics:** Dynamic parallelism, Unified Virtual Memory, Multi-GPU processing, Peer access, Heterogeneous processing. Case Studies: Image Processing, Graph algorithms, Simulations, Deep Learning.	7
	TOTAL	42

REFERENCES

S.No.	Name of Books/Authors/Publishers	Year of Publication / Reprint
1	Wen-mei W. Hwu, David B. Kirk, Izzat El Hajj, "Programming Massively Parallel Processors: A Hands-on Approach", Morgan Kaufmann, 4th Edition.	**2022**
2	Shane Cook, "CUDA Programming: A Developer's Guide to Parallel Computing with GPUs (Applications of Gpu Computing)", Morgan Kaufmann Publishers In, Illustrated edition.	**2012**