Course Code	CSE560
Course Name	GPU Computing
Credits	4
Course Offered to	UG/PG
Course Description	This course will introduce parallel computing paradigms with focus on GPGPU programming to harness the massively parallel GPU architecture in solving computationally demanding tasks. The NVIDIA CUDA and industry standard OpenCL frameworks will be introduced and used with most of the labs. This is a project based course where the students will work on scientific computational problems.

Pre-requisites

Pre-requisite (Mandatory)	Pre-requisite (Desirable)	Pre-requiste (Other)
CSE101 Intro to Programming	CSE102 Data Structures & Algorithms	C/C++ Programming (students must know how to write reasonable length C programs before this course).

*Please insert more rows if required

Post Conditions*(For suggestions on verbs please refer the second sheet)

CO1	**CO2**	**CO3**	**CO4**
Students are able to understand concepts behind parallel computing	Students are able to understand parallel computing paradigms, GPU architecture and GPGPU development frameworks (CUDA, OpenCL, and GLSL)	Students are able to analyse an algorithms to provide parallel solutions to computationally challenging problems	Students are able to implement such solutions on GPU using CUDA, and show effectiveness of the GPU based solutions using standard benchmarks and tools

Weekly Lecture Plan

Week Number	Lecture Topic	COs Met	Assignment/Labs/Tutorial
1	**Introduction and overview**: advances in architecture and technology, need for parallel computing, examples, and challenges.	C01, C02	Weekly programming assignments and homeworks to implement and analyse data structures covered in class. Homeworks contain both theoretical and programming problems.
2,3	**Basics on architecture and programming**: CPU/GPU architecture, multicore architecture, Flynn's taxonomy, SIMT execution model	C01, C02	
4,5	**Introduction to CUDA C**: kernel based data parallel execution model, memory model and locality, CUDA threads, atomics, GPU utilisation	C02	

6,7	**Parallel programming paradigms**: parallel algorithm design, analytical modelling of parallel programs, limits on achievable performance, Amdahl's law, Gustafson's law, scalability, work optimality, message passing, shared address space machines, basic communication operations, concurrency	C01, C03	
8-10	**Parallel computing using CUDA**: data transfer and CUDA streams, performance considerations, floating-point accuracy, synchronisation, communication, reduction trees, parallel prefix sum, optimisations	C01, C02	
11-12	OpenMP, OpenACC, Multi-GPU systems, GPGPU-computing using OpenCL and OpenGL	C04	
13	Case studies	C03	

*Please insert more rows if required

Weekly Lab Plan

Week Number	**Laboratory Exercise**	**COs Met**	**Platform (Hardware/Software)**
General Plan	Getting familiar with the programming environment	C04	
Week 1	Writing basic CUDA C progam - vector addition	C01, C04	CUDA C
Week 2-3	Matrix multiplication in CUDA	C01, C04	CUDA C
Week 4	Matrix multiplication in CUDA using shared memory and memory coalescing	C01, C04	CUDA C
Week 5	Assignment evaluation		CUDA C
Week 6	Project discussions		
Week 7	Assignment evaluation		
Week 8	GPU performance tools, CUDA debugger	C04	CUDA C
Week 9	Assignment evaluation		
Week 10	Project evaluations	C03	CUDA C
Week 11	Project evaluations	C03	CUDA C
Week 12	Project evaluations	C03	CUDA C
Week 13	Project evaluations	C03	CUDA C

*Please insert more rows if required

Assessment Plan

Type of Evaluation	**% Contribution in Grade**
Mid-sem	15
End-sem	30
Project	30
Assignment	20
Quiz	5

Mid sem= theory
End sem= theory

*Please insert more row for other type of Evaluation

Resource Material	
Type	Title
Textbook	David B. Kirk, and Wen-mei W. Hwu, Programming massively parallel processors: a hands-on approach, Elsevier.
Textbook	A. Grama, A. Gupta, G. Karypis, and V. Kumar, Introduction to parallel computing, 2nd edition.