## DISCIPLINE SPECIFIC CORE COURSE– 18 (DSC-18): Introduction to Parallel Programming

**Credit distribution, Eligibility and Pre-requisites of the Course**

| Course title & Code | Credits | Credit distribution of the course | | | Eligibility criteria | Pre-requisite of the course |
|---|---|---|---|---|---|---|
| | | Lecture | Tutorial | Practical/ Practice | | |
| **DSC-18 Introduction to Parallel Programming** | **4** | **3** | **0** | **1** | Pass in Class XII | DSC-02 Computer Systems Architecture DSC04 /A course in C/C++ at plus 2 level, DSC-07 Data Structures with C++, DSC-08 Operating Systems |

**Learning Objectives**

The course introduces the students to the basic concepts and techniques of parallel programming. It enables them to design and implement parallel algorithms. The course would give the students hands-on practice to write parallel programs using shared and distributed memory models using OpenMP and Message Passing Interface (MPI).

**Learning outcomes**

On successful completion of this course, the student will be able to:

- State the need of Parallel algorithms
- Describe architectures for parallel and distributed systems.
- Develop elementary parallel algorithms for shared memory models.
- Develop elementary parallel algorithms for distributed memory models.

**SYLLABUS OF DSC-18**

**Unit 1 (6 Hours)**

**Introduction to Parallel Computing**: Trends in microprocessor architectures, memory system performance, dichotomy of parallel computing platforms, physical organization of parallel platforms, communication costs in parallel machines, SIMD versus MIMD architectures, shared versus distributed memory, PRAM shared-memory model, distributed-memory model.

**Unit 2 (15 Hours)**

**OpenMP programming for shared memory systems**: Thread Basics, Controlling Thread and Synchronization Attributes, Multi-thread and multi-tasking, Context Switching, Basic OpenMP thread functions, Shared Memory Consistency Models and the Sequential Consistency Model, Race Conditions, Scoping variables, work-sharing constructs, critical sections, atomic operations, locks, OpenMP tasks, Introduction to tasks, Task queues and task execution, Accessing variables in tasks, Completion of tasks and scoping variables in tasks, Recursive task spawning and pitfalls

**Unit 3 (15 Hours)**
**MPI programming for distributed memory systems**: MPI basic communication routines (Introduction to MPI and basic calls, MPI calls to send and receive data, MPI call for broadcasting data, MPI Non-blocking calls, MPI Collectives (MPI Collectives and MPI broadcast, MPI Gathering and scattering collectives, MPI reduction and Alltoall collectives, MPI collectives design), Types of interconnects (Characterization of interconnects, Linear arrays, 2D mesh and torus, cliques)

**Unit 4** (9 Hours)
**Applications**: Matrix-matrix multiply, Odd-Even sorting, distributed histogram, Breadth First search, Dijkstra's algorithm

**Essential/recommended readings**


1. Grama, A., Gupta, A., Karypis, G., Kumar, V., *Introduction to Parallel Computing*, 2nd edition, Addison-Wesley, 2003.
2. Quinn, M. *Parallel Programming in C with MPI and OpenMP*, 1st Edition, McGraw-Hill, 2017.
3. Revdikar, L., Mittal, A., Sharma, A., Gupta, S., *A Naïve Breadth First Search Approach Incorporating Parallel Processing Technique For Optimal Network Traversal*, International Journal of Advanced Research in Computer and Communication Engineering Vol. 5, Issue 5, May 2016.


**Additional References**

**No additional references mentioned**

**Suggested Practical List :**

**Practical exercises such as**

1. Implement Matrix-Matrix Multiplication in parallel using OpenMP
2. Implement distributed histogram Sorting in parallel using OpenMP
3. Implement Breadth First Search in parallel using OpenMP
4. Implement Dijkstra's Algorithm in parallel using OpenMP