IV Year – II Semester

L	T	P	C
4	0	0	3

CONCURRENT AND PARALLEL PROGRAMMING

(Elective - 3)

OBJECTIVES:

- Improvement of students comprehension of CPP, new programming concepts, paradigms and idioms
- Change of 'mood' regarding Concurrency counter-intuitiveness
- Proactive attitude: theoretical teaching shouldn't be so dull
- Multipath, individually paced, stop–and–replay, personalized learning process
- Frequent assessment of learning advances on the subject

UNIT- 1

Concurrent versus sequential programming. Concurrent programming constructs and race condition. Synchronization primitives.

UNIT-II

Processes and threads. Interprocess communication. Livelock and deadlocks, starvation, and deadlock prevention. Issues and challenges in concurrent programming paradigm and current trends.

UNIT-III

Parallel algorithms – sorting, ranking, searching, traversals, prefix sum etc.,

UNIT- IV

Parallel programming paradigms – Data parallel, Task parallel, Shared memory and message passing, Parallel Architectures, GPGPU, pthreads, STM,

UNIT-V

OpenMP, OpenCL, Cilk++, Intel TBB, CUDA

UNIT-VI

Heterogeneous Computing: C++AMP, OpenCL

OUTCOMES:

- Understanding improvement of CPP concepts presented
- The number of reinforcement–exercises assigned
- The time required for the resolution of exercises
- Compliance level with the new model of theoretical teaching

TEXT BOOKS:

1. Mordechai Ben-Ari. Principles of Concurrent and Distributed Programming, Prentice-Hall International.
2. Greg Andrews. Concurrent Programming: Principles and Practice, Addison Wesley.
3. GadiTaubenfeld. Synchronization Algorithms and Concurrent Programming, Pearson.
4. M. Ben-Ari. Principles of Concurrent Programming, Prentice Hall.
5. Fred B. Schneider. On Concurrent Programming, Springer.
6. Brinch Hansen. The Origins of Concurrent Programming: From Semaphor