First Course Handout

Course Title: Analysis of Concurrent Programs
Course No: CS 636
Credits: 3-0-0-0-[9]

Prerequisite:

- Exposure to CS 220 (Computer Organization) and CS 330 (Operating Systems) (or equivalent non-IITK courses) is desirable.
- Programming maturity (primarily C/C++/Java) is desirable.

Lecture Hours: Mon 11:00-12:00 PM ThuFri 10:00-11:00 AM in KD 102

Course Objective: This course will focus on the challenges of developing correct and scalable concurrent programs. We will discuss the errors we make while writing concurrent programs and the techniques and tools to debug them. We will learn about techniques and abstractions that help write correct concurrent programs, such as synchronization primitives and transactional memory. Furthermore, we will discuss testing of concurrent programs and ideas to avoid common performance bottlenecks.

The course may include programming assignments and a course project.

Course Contents: The course will primarily focus on the following topics.

- Concurrency Bugs: data races, atomicity violations, and deadlocks
- Shared Memory Synchronization: locks, monitors, semaphores, flags, barriers, condition variables
- Concurrent programming paradigms: shared-memory, message-passing, partitioned global address space
- Memory Consistency Models
- Transactional Memory
- Concurrent Data Structures
- Performance Challenges: PAPI counters, performance analysis tools, false sharing
- Testing of Concurrent Programs

We may add new, drop existing, or reorder topics depending on progress and class feedback.

The course may also involve reading and critiquing related research papers.

The recommended reading material is part of the course.

Policies:

- Please be on time for class.
- Please try to avoid using laptops and/or mobile devices in class, they are distracting for both the instructor and other students who want to concentrate.
- You are allowed up to **two late** days to submit your assignment (if applicable), with a 25% penalty for each day.

Evaluation:

Class participation (e.g., interaction and quizzes)	5%
Assignments (2-3)	15%
Midsem	25%
Term Project (with periodic evaluation)	25%
Endsem	30%

- This is a tentative allocation
    - Might change allocations depending on the strength of the class
- Grading will be relative

References:

1. The Art of Multiprocessor Programming, 2nd edition - M. Herlihy et al.
2. A Primer on Memory Consistency and Cache Coherence, 2nd edition - V. Nagarajan et al.
3. Shared-Memory Synchronization, 2nd edition - M. Scott and T. Brown
4. Transactional Memory, 2nd edition - Tim Harris et al.
5. Parallel Computer Architecture: A Hardware/Software Approach - D. Culler et al.
6. Java Concurrency in Practice - B. Goetz et al.

We may read and discuss related materials and research papers, which we will announce in class.