Course Title	Fundamental of Distributed Systems
Course Code	SEng7431
CP	5 (2hrs Lecture, 3hrs Laboratory)
Module Number	07
Pre-requisites	None
Year	III
Semester	I
Status of Course	Compulsory

This course is an introductory course in distributed systems. The emphasis will be on the techniques for creating functional, usable, and high-performance distributed systems. To make the issues more concrete, the class includes several multi-week projects requiring significant design and implementation.

Learning Outcomes

After this course, students will have learned to:

- Implement and structure distributed systems programs.
- Write programs that can interoperate using well-defined protocols.
- Debug highly concurrent code that spans multiple programs running on multiple cores andmachines.
- Reason about distributed algorithms for locking, synchronization and concurrency, scheduling,and replication.
- Use standard network communication primitives such as UDP and TCP.
- Understand the general properties of networked communication necessary for distributed systemsprogramming in clusters and on the Internet.
- Employ and create common paradigms for easing the task of distributed systems programming, such as distributed file systems, RPC, and Map Reduce. Be able to clearly elucidate their benefits,drawbacks, and limitations.
- Identify the security challenges faced by distributed systems programs.
- Be able to select appropriate security solutions to meet the needs of commonly encountered distributed programming scenarios.

Week	Topic
1-2	**Chapter 1: Introduction to distributed system** definition characteristics organization and goals of distributed systems hardware and software concepts the client-server model
3-4	**Chapter 2: Communication** layered protocols client-server TCP middleware protocols remote procedure call and remote object invocation message oriented and stream oriented communication Quality of Service
5-6	**Chapter 3: Processes** threads and their implementation clients and servers and design issues object servers and adaptors code migration software agents and agent technology agent communication languages
7-8	**Chapter 4: Naming** naming entities name spaces and name resolution; DNS and X.500 different approaches in locating mobile entities identifying and removing (unreferenced) unreachable entities

9-10	**Chapter 5: Synchronization** clock synchronization, physical clocks and clock synchronization algorithms logical clocks and time stamps global state distributed transactions and concurrency control election algorithms mutual exclusion and various algorithms to achieve mutual Exclusion
11-13	**Chapter 6: Consistency and Replication** reasons for replication and object replication replication as scaling technique data-centric and client-centric consistency models distribution and consistency protocols implementation example: Orca
14-16	**Chapter 7: Fault Tolerance** basic concepts failure modes failure masking by redundancy process resilience reliable client-server and group communication distributed commit recovery

Summary of Teaching Learning Methods

The course will be delivered in the form of lectures, demonstration, laboratory sessions, groupDiscussions, and individual and group project works.

Assessment Methods: - As per the academic regulation

References

1. K. Birman, Building Secure and Reliable Network Applications, Manning PublicationsCo., 1996
2. S. Tanenbaum and Maarten van Steen, Distributed Systems, Principles and Paradigms,Prentice Hall, 2002
3. S. Mullender, Distributed Systems, 2nd edition, Addison-Wesley, 1993