राष्ट्रीय प्रौद्योगिकी संस्थान पटना / **NATIONAL INSTITUTE OF TECHNOLOGY PATNA**
(शिक्षा मंत्रालय, भारत सरकार के अधीन एक राष्ट्रीय महत्व का संस्थान / An Institute of National Importance under Ministry of Education, Gov. of India)
संगणक विज्ञान एवं अभियांत्रिकी विभाग / **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**
अशोक राजपथ, पटना - ८००००५, बिहार / Ashok Rajpath, Patna- 800005, Bihar
Tel. No. – 0612-2372715, 2370419 (Ext-200)   email- cseoffice@nitp.ac.in

*CSXX02*50: **Distributed Computing**

L-T-P-Cr: 3-0-0-3

Prerequisites: Computer Organization, Operating Systems, Computer Networks.

Course Overview: The purpose of this course is to cover essential concepts of distributed computing (particularly, fundamental principles of distributed computing and middleware platforms, as well as major paradigms of distributed computing), and to relate these concepts to contemporary distributed computing systems.

Course Outcomes: After completing this course, students should be able to:
- CO-1. *recall* distributed computing terminologies, and fundamental concepts, operational principles and methods of its communication, synchronization, consistency, replication and failure recovery functionalities;
  [Bloom level: *Remember*; Mapped to: PO-1, PSO-1]
- CO-2. *recognize* communication paradigms and architectural patterns in distributed computing;
  [Bloom level: *Remember*; Mapped to: PO-1, PSO-1]
- CO-3. *explain* concepts and techniques for functional/nonfunctional characteristics of architectural elements and patterns in architectural models of distributed computing;
  [Bloom level: *Understand*; Mapped to: PO-1, PO-2, PSO-1]
- CO-4. *solve* problems on principles of multithreading, communication, naming, clock synchronization, conit, replication and checkpointing, and their performances in distributed computing;
  [Bloom level: *Apply*; Mapped to: PO-1, PO-2, PO-3, PSO-1]
- CO-5. *determine* nature, correctness and complexity of synchronization, coordination, consistency, consensus and availability that are supported in distributed computing;
  [Bloom level: *Analyze*; Mapped to: PO-1, PO-2, PO-3, PSO-1]
- CO-6. *evaluate* protocols of communication, agreement, commit, consensus and allocation in given distributed computing systems.
  [Bloom level: *Evaluate*; Mapped to: PO-1, PO-2, PO-4, PO-5, PSO-1]

Mapping of COs to POs and PSOs (Course Articulation Matrix):

COs	Bloom levels	POs												PSOs	
		PO-1 (Engineering knowledge)	PO-2 (Problem analysis)	PO-3 (Design/ development of solutions)	PO-4 (Conduct investigations of complex problems)	PO-5 (Modern tool usage)	PO-6 (The engineer and society)	PO-7 (Environment and sustainability)	PO-8 (Ethics)	PO-9 (Individual and team work)	PO-10 (Communication)	PO-11 (Project management and finance)	PO-12 (Life-long learning)	PSO-1 (Complex software & info mgmt. systems)	PSO-2 (Sustainable intelligent, secure computing solutions)
CO-1	Remember	3												3	
CO-2	Remember	3												3	

COs	Bloom levels	POs												PSOs	
		PO-1 (Engineering knowledge)	PO-2 (Problem analysis)	PO-3 (Design/ development of solutions)	PO-4 (Conduct investigations of complex problems)	PO-5 (Modern tool usage)	PO-6 (The engineer and society)	PO-7 (Environment and sustainability)	PO-8 (Ethics)	PO-9 (Individual and team work)	PO-10 (Communication)	PO-11 (Project management and finance)	PO-12 (Life-long learning)	PSO-1 (Complex software & info mgmt. systems)	PSO-2 (Sustainable intelligent, secure computing solutions)
CO-3	Understand	3	3											3	
CO-4	Apply	3	3	3										3	
CO-5	Analyze	3	3	3										3	
CO-6	Evaluate	3	3		3	3								3	

Note: Strong relation = 3; Moderate relation = 2; Weak relation = 1; No relation = 0.

Syllabus:

Unit I   Lectures: 17

(Module-1) *Introduction*: Motivation for distributed computing; Operating system support; Middleware categories; Network service requirements; Network protocol stack support.

(Module-2) *Architectural model*: Architectural models of distributed computing; Communicating entity; Entity placement; Communication paradigm; Remote invocation, indirect communication; Architectural pattern; Heterogeneity, transparency, concurrency, resiliency; Portability, interoperability, scalability.

(Module-3) *Communication paradigm*: Multithreading in distributed computing; Virtualization; Server organization, distribution; Address rewriting; Redirection; Code mobility; Process-resource-machine binding; Code-on-demand, mobile agent; Migration; Virtual machine migration; Decoupling; Distributed shared memory; Message passing; Remote procedure call; Client/server; Group communication; Publish-subscribe paradigm.

Unit II   Lectures: 10

(Module-1) *Naming*: Naming scheme; Namespace; Uniform Resource Identifier; Name resolution; Name service; Naming mobile entities; Location service; Distributed garbage collection.

(Module-2) *Synchronization*: Clock synchronization; NTP; Wireless clock synchronization; Logical clock; Happen-before relation; Lamport's clock; Vector clock; Agreement; Reliable multicasting; Election model; Distributed mutual exclusion.

(Module-3) *Coordination*: Data-driven, control-driven coordination; Gossip; Epidemic model; Group coordination; Publish-subscribe coordination; Event service; Overlay.

Unit III   Lectures: 15

(Module-1) *Consistency*: Consistency model; Data-centric consistency; Linearizability; Sequential, causal, FIFO consistencies; Eventual consistency; Client-centric consistency.

(Module-2) *Replication*: Active, passive replication; View-synchronous group; Consistent replication; Conit; Transparent replication; Dynamic migration/replication; Push, pull, lease; Replication-aware invocation.

(Module-3) *Fault tolerance*: Failure, error, fault; Fault-to-failure transition; Fault tolerance; Dependability, availability, reliability; Failure model; CAP theorem; Masking failure; Fault-tolerant process group; Voting quorum; Well-defined failure behavior; Distributed commit; Fault tolerance agreement; Byzantine agreement; Consensus; Paxos consensus.

(Module-4) *Failure recovery*: Forward, backward failure recovery; Local, global states; Distributed snapshots; Checkpoint; Rollback recovery; Always-no-orphans; Pessimistic, optimistic, causal message logging.

Text/Reference Books:

1. Ajay D. Kshemkalyani and Mukesh Singhal, *Distributed Computing: Principles, Algorithms, and Systems*, Cambridge University Press, 2008 (Indian Reprint, 2010, ISBN: 9781107648906).
2. Maarten van Steen and Andrew S. Tanenbaum, *Distributed Systems*, Fourth edition, 2023 (ISBN: 9789081540636).
3. George Coulouris, Jean Dollimore, Tim Kindberg and Gordon Blair, *Distributed Systems: Concepts and Design*, Pearson/Addison-Wesley, Fifth edition, 2012 (Pearson Education Indian Reprint, 2017, ISBN: 9789332575226).
4. Joel M. Crichlow, *Distributed Systems: Computing Over Networks*, Second edition, Pearson/Prentice-Hall, 2014 (PHI Indian Reprint, 2014, ISBN: 9788120348776).
5. Andrew S. Tanenbaum and Maarten van Steen, *Distributed Systems: Principles and Paradigms*, Pearson/Prentice-Hall, Second edition, 2007 (Pearson Education Indian Reprint, 2015, ISBN: 9789332549807).
6. Pankaj Jalote, *Fault Tolerance in Distributed Systems*, Prentice-Hall, 1994 (ISBN: 9780133013672).
7. Mukesh Singhal and Niranjan G. Shivaratri, *Advanced Concepts in Operating Systems*, McGraw-Hill, 1994 (Tata McGraw-Hill Indian Reprint, 2008, ISBN: 9780070472686).
8. Sukumar Ghosh, *Distributed Systems: An Algorithmic Approach*, Chapman and Hall/CRC, Second edition, 2014 (ISBN: 9781138570009).
9. Arno Puder, Kay Römer and Frank Pilhofer, *Distributed Systems Architecture: A Middleware Approach*, Elsevier, 2006 (Indian Reprint, ISBN: 9788131216651).