

**SUBJECT DESCRIPTION FORM**

---

Subject Title: Distributed Computing

---

Subject Code: COMP5325

---

Credit Value: 3

---

Pre-requisite: (Subject title and code no, if any)

Nil

---

Recommended background knowledge:

Programming skills in C or C++; knowledge in Unix.  
Background in Operating Systems is strongly recommended.

---

Mutual Exclusions:

Distributed Computing (COMP534)

---

Learning Approach:

42 hours of class activities including - lecture, tutorial, lab, workshop seminar where applicable

---

Assessment:

Continuous assessment	45%
Test, and Examination	55%

---

Objectives:

To provide in-depth study in the area of distributed computing on models, architectures, algorithms and techniques and to allow the student to:

- acquire fundamental knowledge in distributed computing;
  - learn about advanced distributed computing concepts;
  - understand limitations and appreciate innovative solutions; and
  - apply the knowledge in distributed application development and problem solving.
- 

*The Department reserves the right to update the syllabus contents. Please note that the learning approach for the same subject could vary slightly due to different delivery modes.*

Learning Outcomes:

After completing the subject, students should be able to:

1. understand, appreciate and apply parallel and distributed algorithms in problem solving;
2. learn advanced techniques and their application in practical systems;
3. evaluate the impact and performance of network topology on parallel / distributed algorithm formulation; and
4. gain hands-on experience with those programming techniques.

Keyword Syllabus:

1. **Overview and background of distributed computing**

Motivations; applications; distributed systems and architecture; computational model; causal dependency; physical versus logical clock; vector clock; distributed snapshot; remote procedure call; client-server interaction; broadcast versus multicast.

2. **Synchronization and coordination**

Distributed synchronization; clock synchronization; mutual exclusion; quorum consensus; leader election; other synchronization problems; deadlock prevention; deadlock detection; load balancing; process migration; fault-tolerance; synchronous and asynchronous checkpointing; recovery.

3. **Shared data access**

Atomic data access; transactions; concurrency control; atomic commitment; distributed file systems; stateful versus stateless server; replicated data management; primary copy approach; distributed shared memory.

4. **Distributed programming**

TCP/IP and sockets; POSIX threads; API; distributed operating systems.

5. **Selected topics on distributed computing**

Advanced or current topics on distributed computing; examples include MPI, DCE, Java applet and servlet, Internet computing, mobile computing.

---

Indicative reading list and references:

Coulouris, G.F., Dollimore, J. and Kindberg, T., 2001, *Distributed Systems: Concepts and Design*, Third Edition, Addison-Wesley.

Singhal, M. and Shivaratri, N.G., 1994, *Advanced Concepts in Operating Systems*, McGraw Hill.

Chow, R. and Johnson, T., 1997, *Distributed Operating Systems and Algorithms*, Addison-Wesley.

Goscinski, A., 1991, *Distributed Operating Systems: The Logical Design*, Addison-Wesley.

Mullender, S., ed., 1993, *Distributed Systems*, Second Edition, Addison-Wesley.

Tanenbaum, A., 1995, *Distributed Operating Systems*, Prentice-Hall.

Articles from journals, magazines, and conference proceedings, including *JACM*, *ACM TOCS*, *ACM TODS*, *ACM TOPLAS*, *IEEE TPDS*, *IEEE TKDE*, *IEEE TSE*, *IEEE TOC*, *CACM*, *IEEE Computer*, *IEEE Internet Computing*, *PODC*, *SOSP*, *SPAA*, *ISCA*, *ICDCS*, *FTCS*, *RDS*, *WWW*.