UNIVERSITY of NICOSIA

Course Syllabus

Course Code	**Course Title**	**ECTS Credits**
COMP-417	Parallel and Distributed Computing	6
Prerequisites	**Department**	**Semester**
COMP-212, COMP-354	Computer Science	Fall
Type of Course	**Field**	**Language of Instruction**
Required	Computer Science	English
Level of Course	**Lecturer(s)**	**Year of Study**
1st Cycle	Dr Harald Gjermundrød	3rd or 4th
Mode of Delivery	**Work Placement**	**Corequisites**
Face-to-face	N/A	None

Course Objectives:

The main objectives of the course are to:

- introduce models, mechanisms, and techniques of parallel and distributed software development
- understand when concurrent programming techniques are appropriate to use and the benefits and risks involved with choosing such an approach
- thoroughly discuss concurrent programming paradigms or patterns like threads, safety, liveliness, state guarding, atomicity, and locking
- cover in detail object visibility and immutability as well as thread safety and confinement
- explain in detail how to go from requirements to model to implementation of parallel and distributed applications
- compare and contrast different message passing paradigms
- cover in detail how to model and develop dynamic systems
- expose the students to development tools/environments/frameworks to develop parallel and distributed systems.

Learning Outcomes:

After completion of the course students are expected to be able to:

1. model, analyze, develop, and verify parallel and distributed systems
2. analyze problems and apply applicable parallel and distributed solutions when developing software

3. correctly apply concurrent concepts like threads, state guarding, atomicity, and locking for developing parallel applications
4. apply techniques for verifying liveness and safety properties in parallel applications
5. compare and contrast the different message passing paradigms that can be used in a parallel application
6. demonstrate how to go from system requirements to a system model, which then is implemented as a software system for real world problems
7. apply techniques to developing dynamic systems
8. develop (write/debug/correct) parallel applications that satisfy safety and liveness properties in various programming languages.

Course Content:

1. Introduction Parallel and Distributed Computing
   a) Modeling computing systems
   b) Shared memory computing systems
   c) Distributed computing systems
2. Concurrent Execution
   a) Modeling concurrency
   b) Processes and threads
   c) Multi-threaded program development
3. Shared Objects and Mutual Exclusion
   a) Interference
   b) Modeling mutual exclusion
   c) Programming mechanisms for mutual exclusion
4. Process Synchronization
   a) Condition synchronization
   b) Locking mechanism like semaphores, monitors
   c) Nested locking issues
   d) Deadlock examples and analysis
5. Liveness and Safety Properties
   a) Modeling liveness and safety
   b) Starvation vs. deadlock

6. Model-based Design
   a) Requirements to models
   b) Models to implementation
7. Dynamic Systems
   a) Fair allocation
   b) Bounded overtaking
   c) Master-slave pattern
8. Message Passing
   a) Synchronous message passing
   b) Asynchronous message passing
   c) Rendezvous
   d) Client-server model

Learning Activities and Teaching Methods:

Lectures, Practical Exercises, Projects, and Assignments.

Assessment Methods:

Final Exam, Midterm Exam, Project, and Quizzes

Required Textbooks / Readings:

Title	**Author(s)**	**Publisher**	**Year**	**ISBN**
Concurrency: State Models and Java Programs, 2nd Edition	Jeff Magee, Jeff Kramer	Wiley	2006	978-0470093559
Topics in Parallel and Distributed Computing	S. Prasad, A. Gupta, A. Rosenberg, A. Sussman, C. Weems	Elsevier	2015	978-0128039380

Recommended Textbooks / Readings:

Title	Author(s)	Publisher	Year	ISBN
Principles of Concurrent and Distributed Programming	M. Ben-Ari	Addison-Wesley	2006	978-0321312839
Java Concurrency in Practice	B. Goetz, T. Peierls, J. Bloch, J. Bowbeer, D. Holmes, D. Lea	Addison-Wesley Professional	2006	978-0321349606