



# ID1217 Concurrent Programming 7.5 credits

## Programmering av parallella system

This is a translation of the Swedish, legally binding, course syllabus.

If the course is discontinued, students may request to be examined during the following two academic years

## Establishment

The official course syllabus is valid from the Spring semester 2024 in accordance with the decision by the Head of School: J-2023-2691. Date of decision: 2023-10-16

## Grading scale

A, B, C, D, E, FX, F

## Education cycle

First cycle

## Main field of study

Information Technology, Technology

## Specific prerequisites

Knowledge of computer technology/computer architecture including multicore architectures, 6 higher education credits, equivalent to a completed course IS1200/IS1500/EP1200/HE1028.

Knowledge and skills in C-programming, 3 higher education credits, equivalent to a completed course IS1200/IS1500 or DD1388/SF2565/HI1024.

Knowledge and skills in object oriented programming, 6 credits, equivalent to completed course DD1310-DD1319/DD1321/DD1331/DD1337/DD100N/ID1018.

## Language of instruction

The language of instruction is specified in the course offering information in the course catalogue.

## Intended learning outcomes

After passing the course, the student should be able to

1. compare and contrast programming models and paradigms for shared memory and distributed memory to choose the most appropriate models and the paradigms for a given problem
2. justify, illustrate and explain the need of synchronisation in a concurrently executing programme
3. explain, compare, choose and use the different synchronisation mechanisms in a concurrently executing programme
4. explain, compare, choose and use different communication mechanisms for process interaction in distributed programmes
5. implement concurrently executing and distributed programmes in C, Java or other programming languages by means of various programming environments such as pthreads, openMP, Java threads and monitors, Socket API, MPI and Java RMI
6. evaluate, measure and estimate speed, performance and scalability and analyse safety and liveness properties for parallel and distributed programmes

in order to

- acquire knowledge in the basics of concurrent programming, more specifically about programming models, concepts, paradigms, techniques, synchronisation and communication mechanisms
- develop practical skills and experience in programming environments for developing parallel and distributed programmes

## Course contents

Parallel Programming with threads and shared variables Processes and synchronisation  
Critical sections, locks, barriers, semaphores and monitors Language overview: threads in Java, Pthreads

Introduction to parallel and distributed programming with processes

Message passing, RPC, RMI and rendezvous

Overview of parallel and distributed programming environments e.g. MPI, PVM and Open-MP Performance

## Examination

- LABA - Laboratory Work, 3.0 credits, grading scale: P, F
- TEN1 - Examination, 4.5 credits, grading scale: A, B, C, D, E, FX, F

Based on recommendation from KTH's coordinator for disabilities, the examiner will decide how to adapt an examination for students with documented disability.

The examiner may apply another examination format when re-examining individual students.

Examination is computer-based and should be made in place at KTH Royal Institute of Technology

## Other requirements for final grade

Passed written examination (TEN1; 4.5 higher education credits) and passed programming assignments (LABA; 3.0 higher education credits)

## Ethical approach

- All members of a group are responsible for the group's work.
- In any assessment, every student shall honestly disclose any help received and sources used.
- In an oral assessment, every student shall be able to present and answer questions about the entire assignment and solution.