# Course PM ID2010 VT20

## Expected learning outcomes

The course aim is that students will be able to analyze, criticize and make design decisions that affects the interaction between components in distributed systems.

This is achieved by being able to:

- recognize and identify characteristic properties in a distributed computer system
- describe and use basic computer communication technology, with an emphasis on TCP/IP
- discuss and reflect over risks and opportunities associated with interactive systems

The above abilities are demonstrated by the student in a closed exam.

- analyze and extend a given distributed application with multiple components and asynchronous interactions
- construct a simpler distributed and interactive systems consisting of components that are autonomous, reactive and proactive

The above abilities are demonstrated by the student in the solution, demonstration and documentation of a programming assignment.

## Contents

The course is conducted in three parallel tracks; lectures (~18 hours), literature studies, and assignments (~ 50 hours) executed in groups of one or two students. Lectures provide the overall framework and the orientation on theory and practice regarding the assignments. This is complemented with technology and case-studies.

The main course literature is used as background and reference material for the Java-based sections. An article collection illuminates additional core issues regarding the agent metaphor and fundamental differences between local and distributed programs.

The Canvas LMS is the primary means of communication and information.

## Course prerequisites

Working knowledge of the Java programming language. Being acquainted with threads, synchronization mechanisms and Remote Method Invocation is a plus.

## Course literature

Java Network Programming and Distributed Computing, David Reilly, Michael Reilly
Upplaga: Förlag: Addison Wesley År: 2002
ISBN: 0-201-71037-4

Students are encouraged to select the relevant material from more recent titles. See the reading guide below.

## A collection of articles:

1. Jim Waldo, Geoff Wyant, Ann Wollrath, Sam Kendall, “A Note on Distributed Computing”, (smli_tr-94-29.pdf).
2. Nicholas A. Jennings, “An agent-based approach for building complex software systems”, (p35-jennings.pdf).
3. Nicholas A. Jennings, “On agent-based software engineering”, section 5: “The downside of an agent-based approach to software engineering”, (Jennings_1999.pdf).
4. Pablo Basanta-Val and Jonathan Stephen Anderson, “Using Real-Time Java in Distributed Systems: Problems and Solutions”
   `https://link.springer.com/chapter/10.1007%2F978-1-4419-8158-5_2`
   `rtsj.org`
   `dds-foundation.org`
5. *Further required or suggested reading may be announced as the course starts.*

## Reading guide to Reilly & Reilly

| | | |
|---|---|---|
| Chapter 1 | Networking Theory | mandatory |
| Chapter 2 | Java Overview | sec. 2.1-2.4 recommended<br>sec. 2.5-2.6 mandatory<br>sec. 2.7-2.9 recommended |
| Chapter 3 | Internet Addressing | recommended |
| Chapter 4 | Data Streams | sec. 4.1-4.4 recommended<br>sec. 4.5-4.6 mandatory |
| Chapter 5 | User Datagram Protocol | sec. 5.1 mandatory<br>sec. 5.2-5.6 recommended |
| Chapter 6 | Transmission Control Protocol | sec. 6.1-6.3 mandatory<br>sec. 6.4 recommended<br>sec 6.5-6.9 mandatory |
| Chapter 7 | Multi-threaded Applications | mandatory |
| Chapter 8 | Implementing Application Protocols | skip |
| Chapter 9 | HyperText Transfer Protocol | skip |
| Chapter 10 | Java Servlets | skip |
| Chapter 11 | Remote Method Invocation | mandatory |
| Chapter 12 | Java IDL and CORBA | mandatory |
| Chapter 13 | JavaMail | skip |

- Mandatory – required reading
- Recommended – read it if this is new to you
- Skip – can be omitted

The book is quite naturally very oriented towards Java and the APIs offered therein. We would therefore like to stress that it is not the APIs themselves that are important to the course. Pay attention instead to the functionality implemented by the APIs and the wider concepts that are likely to be independent of the programming language. The book can also in most cases safely be substituted with other titles that cover the same topics.

When reading chapter 7 and 11, compare with the article in the collection: “A note on distributed computing”, by Waldo, Wyant, Wollrath and Kendall.

Further resources and study material may be announced during the course. Consult the Canvas course pages for all online material.

## Examination

- TEN1 – closed exam, no aids allowed, A-F, 4.5hp
- LAB1 – programming assignment, P/F, 3hp

## Grading criteria

The programming assignment is graded P/F (pass/fail). When grading an assignment the criteria below apply individually to the group members.

The grade F (fail) must be given

- if the programming contribution is missing or without substance, or
- if a lab report is missing or without substance

For a passing grade it is required that the chosen extensions to the given systems are completely implemented and demonstrable. The report clearly shows the work and its design decisions. There is a clear connection between the solutions and their execution.

Both parts of the programming assignment must be completed for a passing grade to be awarded.

## Written exam

The design of the written exam may vary between different instances of the course, and from this it follows that the assessment also is adapted. The purpose of the written exam is to provide the student with an opportunity to account for his or hers relation to the learning outcomes, and to do this in a way that makes it possible to merit both width and depth. For this reason the exam is designed to accommodate a grading which requires an observable minimum performance in terms of width, in order to achieve at least a grade of E. From this base, the grading criteria merits such qualities as the understanding of complicated relationships, the ability to reflect, synthesis and insights beyond quotes from the literature. These are interpreted as indications that the learning outcomes has been met in a way that motivates a higher grade.

Note that the width criteria does not imply that the exam must ask about each and every element in the course. Instead a few (3-5) examination subjects are defined for the instance of the exam. The subjects should, when combined with the assignments, cover all the learning outcomes on the course. From the subjects samples are taken that form tasks or questions in the exam.

For the whole exam the grading scale A/B/C/D/E/Fx/F is used. The assessment of individual questions depends on the exam instance. However, the weighting into a summative exam grade is performed according to these criteria:

The grade F must be given if the response contains one or more incomplete examination subjects. If only one subject is incomplete, the deficiencies must be so severe that Fx is not an alternative.

The grade Fx must be given if the response contains at most one incomplete examination subject, which is estimated to be possible to complete by additional examination.

For a grade of E it is required that all examination subjects have been given acceptable responses. The responses reflects the literature in a correct but yet mechanical way.

For a grade of D it is required that most subjects have been given acceptable responses, but one or two indicates deeper understanding and insights.

For a grade of C it is required that all subjects are well responded to and demonstrates deeper understanding and insights. The response to a single subject reflects on the technology and its use, or contains relations and draws conclusions.

For a grade of B it is required that all subjects are well responded to (according to C) and some (but not all) are outstandingly well responded to (according to A).

For a grade of A it is required that all subjects are outstandingly well responded to. The responses reflect on the technology and its use, contains relations and draws conclusions.

## Course grade

The summative course grade is the grade of the written exam (tenta), on the condition that the programming assignment has been given a passing grade. If either or both of the written exam and the programming assignment have been given the grade F, the course grade is F.

## Course coordinator (kursansvarig) and teacher

Fredrik Kilander, fki@kth.se