UNIVERSITY OF GOTHENBURG

# DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

## DIT670 Computer Networks, 7.5 credits

Datornätverk, 7,5 högskolepoäng

*Second Cycle*

---

### Confirmation

This course syllabus was confirmed by Department of Computer Science and Engineering on 2020-11-06 and was last revised on 2020-11-06 to be valid from 2021-08-30, autumn semester of 2021.

*Field of education:* Science 100%
*Department:* Department of Computer Science and Engineering

### Position in the educational system

The course is part of the Computer Science Master's Programme. It is also a single subject course at the University of Gothenburg.

The course can be part of the following programme: 1) Computer Science, Master's Programme (N2COS)

| *Main field of studies* | *Specialization* |
|---|---|
| Computer Science | A1F, Second cycle, has second-cycle course/s as entry requirements |

### Entry requirements

The requirement for the course is to have a Bachelor´s degree within the subject Computer Science or equivalent.

The student should also have successfully completed the courses listed below, either as part of the Bachelor´s degree or as single subject courses.

- 7.5 hec in discrete mathematics (DIT980 or equivalent),
- 7.5 hec in finite automata theory and formal language (DIT322 or equivalent),
- 7.5 hec in imperative or object oriented programming (DIT012, DIT952, or equivalent),

- 7.5 hec in data structures (DIT961 or equivalent),
- 7.5 hec in algorithms (DIT602 or equivalent), and
- 7.5 hec in computer communication (DIT423 or equivalent) or operating systems (DIT401 or equivalent).

Applicants must prove knowledge of English: English 6/English B or the equivalent level of an internationally recognized test, for example TOEFL, IELTS.

**Learning outcomes**

On successful completion of the course the student will be able to:

*Knowledge and understanding*

- demonstrate a broad knowledge of Internet technology and domain name systems,
- demonstrate a considerable degree of technical knowhow on the Internet new generation IPv6,
- describe and analyse architecture, core protocols, global routing, services as well as their limitations of networks such as the Internet,
- discuss and analyse contemporary networking problems, such as TCP connections, contention, performance and flow control,

*Competence and skills*

- define systematically and analyse a computer network in terms of communication graphs and as a distributed system. This specifically refers to problems such as token circulation, spanning tree construction, leader election, initialization of data link algorithms, topology update, clock synchronization, and more,
- use a number of proof techniques, such as re-computation of floating output, fair composition, variant functions, and convergence stairs, as well as demonstrate the correctness of a number of fault-containment or super-stabilization algorithms,
- develop small scale network applications using fundamental networking techniques,
- design and develop your own network-oriented program and then test and demonstrate it in the lab,
- demonstrate and write lab reports on protocol correctness,
- explain and demonstrate the correctness of the studied (self-stabilizing) protocols as well as clearly describe the network algorithms that you design yourself,
- demonstrate software developments for advanced fault-tolerant client-server and peer-to-peer architectures,
- design distributed (self-stabilizing) algorithms for computer networks and show why they work,

*Judgement and approach*

- describe, design and analyse existing and new algorithms for network protocols with a very strong emphasis on self-stabilizing algorithms for computer networks,
- critically analyse the effect of failures, such as transient faults, message omission, and topology changes, on the system and how can such failures propagate and effect computer networks.

**Course content**

This course focuses on the algorithmic design of network protocols and covers a range of sub-specialties including: computer communication network concepts, programming using BSD socket API, and distributed fault-tolerance algorithms with a very strong emphasis on self-stabilization. The aim of this course is to learn to design and analyse algorithms for network protocols and to gain knowledge in existing communication networks, including supporting systems and protocols, fundamental tasks and methods in data communication networks. Mastery of computer networks involves both theory and practice in the design, implementation and use of network protocols and services.

This course offers learning experiences that involve hands-on experimentation and analysis as they reinforce student understanding of concepts and their application to real-world problems. Several laboratory experiments are given and involve API programming for fault-tolerance network systems, and Internet interconnections and services from a practical perspective, and design and analysis of protocols with a strong emphasis on self-stabilizing algorithms.

This course provides the students with the analytical background needed for understanding fundamental issues in the design of distributed fault tolerance algorithms for computer network protocols. The course has a very strong emphasis on transient faults and how self-stabilizing algorithms can allow automatic recovery after their occurrence.

*Sub-courses*

1. **Written examination** *(Skriftlig tentamen),* 6 credits
   Grading scale: Pass with Distinction (VG), Pass (G) and Fail (U)

2. **Assignments** *(Inlämningsuppgifter),* 1.5 credits
   Grading scale: Pass (G) and Fail (U)

**Form of teaching**

Lectures, exercises, home assignments, and laboratory assignments.

*Language of instruction:* English

**Assessment**

The course is examined by an individual written examination carried out in an examination hall at the end of the course, and written assignments which are normally carried out in pairs.

If a student, who has failed the same examined component twice, wishes to change examiner before the next examination, a written application shall be sent to the department responsible for the course and shall be granted unless there are special reasons to the contrary (Chapter 6, Section 22 of Higher Education Ordinance).

In cases where a course has been discontinued or has undergone major changes, the student shall normally be guaranteed at least three examination occasions (including the ordinary examination) during a period of at least one year from the last time the course was given.

**Grades**

The grading scale comprises: Pass with Distinction (VG), Pass (G) and Fail (U).
In order to be awarded Pass (G) for a full course, the student must Pass both the written examination at the end of the course and get approval for all assignments (pre-/post-lecture questions, algorithm assignments, programming labs, and networking labs). To be awarded Pass with Distinction (VG) for a full course, the student must get the grade Pass with Distinction on the written examination and the grade Pass on the assignments.

**Course evaluation**

The course is evaluated through meetings both during and after the course between teachers and student representatives. Further, an anonymous questionnaire is used to ensure written information. The outcome of the evaluations serves to improve the course by indicating which parts could be added, improved, changed or removed.

**Additional information**

This is a work intensive course with lots of homework, labs and reading to do before and after each lectures.

The course is a joint course together with Chalmers.

Course literature to be announced the latest 8 weeks prior to the start of the course.

The course replaces the course DIT665 Computer Networks, 7.5 hec. The course cannot be included in a degree which contains DIT665. Neither can the course be included in a degree which is based on another degree in which the course DIT665 is included.