Masters Programme in Computer Science

Choosing Tracks

Masters Programme

Studies on the advanced level

Acquire specialised knowledge, skills and capacities in relation to first-cycle programmes by:

- further developing the students' ability to integrate and apply knowledge autonomously
- developing the students' ability to deal with complex phenomena, issues and situations
- developing the students' potential for employment that demands great autonomy or for research and development work

Role of the Tracks

Students should

- demonstrate knowledge and understanding in the main field of study, including both broad knowledge of the field and a considerable degree of specialised knowledge in certain areas of the field as well as insight into current research and development work
- demonstrate specialised methodological knowledge in the main field of study
- demonstrate the skills required for participation in research and development work or autonomous employment in some other qualified capacity.

Words of advice

- Design your track to go deep into one field of CS
- Make use of excellent research at KTH
- Select track courses around a common theme

Tracks

- Autonomous Systems
- Computer Security
- IT Management
- Program System Technology
- Language Technology
- Theoretical Computer Science
- Computational Biology
- Computer Systems Engineering
- Sound and Music Computing

Autonomous Systems

- Robotics
- Artificial Neural Networks
- Search Engines
- Image Processing and Computer Vision
- Machine Learning
- Pattern Recognition
- Distributed AI

Computer Security

- Software Security
- Network Security
- Basic Cryptography
- IT and the Law
- Parallel and Distributed Processing

IT Management

- IT-Management with Enterprise Architecture
- Global IT Management
- Industrial Economy
- Project Management
- Process Development and Quality
- Decision Theory

Program System Technology

- Parallel and Distributed Processing
- Automata and Language
- Advanced Object Oriented Systems
- Internet Programming
- Software Security
- Compiler Construction
- Problem Solving and Programming under Stress

Language Technology

- Language Technology
- Multi-modal Interactions and Interfaces
- Search Engines and Information Search Methods
- Automata and Language
- Machine Learning
- Internet Programming

Theoretical Computer Science

- Basic Cryptography
- Complexity Theory
- Statistical Methods in Computer Science
- Automata and Language
- Combinatorics
- Parallel and Distributed Processing
- Software Security

Computational Biology

- Algorithmic Bioinformatics
- Omics and Systems Biology
- Artificial Neural Networks
- Machine Learning
- Neural- and Biomodelling
- Genetics and Genomics
- Cell- and Molecular Biology

Computer Systems Engineering

- Compilers and Execution Environments
- Parallel Computer Systems
- Software for Embedded Systems
- Computer Systems Architectures
- Digital Construction

Sound and Music Computing

- Audio Techniques
- Music Acoustics
- Sound in Interaction
- Multi-modal Interaction and Interfaces
- Speech Technology
- Recognition of Speech and Speakers