



# ID1018 Programming I 7.5 credits

## Programmering I

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 autumn semester 2022 in accordance with the decision from the head of school: J-2022-0550. Decision date: 2022-03-22.

## Grading scale

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

## Education cycle

First cycle

## Main field of study

Technology

## Specific prerequisites

## 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 students should be able to:

- Analyse, explain, modify and increase a simple program in terms of fundamental program constructs and concepts.
- Describe how numerical and sign based data are represented and discuss the use of primitive data types and built-in data structures.
- Design, implement, test and troubleshoot a programme that uses basic calculation simple I/O, common conditionals and iteration structures and function definitions.
- Discuss the importance of algorithms, identify the necessary properties of good algorithms and create algorithms to solve simple problems.
- Develop code that handles exceptional states during the execution.
- Defend the philosophy behind object-oriented design and the concepts encapsulation, abstraction, inheritance and polymorphism.
- Design, implement, test and troubleshoot simple programmes in an object-oriented programming language.

## Course contents

Fundamental designs:

- Basic syntax and semantics of a high level language
- Variables, data types, expressions and assignment
- Simple I/O
- Conditional and iterative control structures
- Functions and parameter transfer
- Structured decomposition

Data structures:

- Representation of numerical data
- Extent, precision and round-off errors
- Arrays
- Representation of character-based data
- Strings and string management
- Memory management during execution
- Pointers and references

Algorithmic problem-solving:

- Problem-solving strategies
- The role of algorithms in the problem solving process
- Implementation strategies for algorithms

- Troubleshooting strategies
- The concepts and properties of algorithms

#### Object-Oriented Programming:

- Object-oriented design
- Encapsulation and concealed information
- Separation of behaviour and implementation
- Classes and subclasses
- Inheritance
- Polymorphism

#### Exceptions

- Exception management

## Examination

- LABB - Programming Assignments, 4.5 credits, grading scale: P, F
- TEN1 - Written Exam, 3.0 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.

Description of the parts of the examination:

LABB – the Student demonstrates creativity and programming skills by completing a definite number of assignments.

TEN1 – the Student demonstrates his ability to analyse and argue with programming concept.

## Transitional regulations

For a student registered in the course for the first time in the autumn semester 2021 or earlier and not having a Pass grade on module LAB1 the following applies: up to the autumn semester 2024, the student after having consulted examiner can choose to become graduate on either LAB1 or LABB, thereafter only on LABB. For a student who has a Pass grade on LAB1, this is taken into account in the final grade.

## 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.