



Åbo Akademi  
University

# Programming Education @ ÅA

**Kristian Nybom**  
Faculty of Science and Engineering  
Information Technologies

5 december 2025



# First year courses

The programming courses taught at Åbo Akademi during the first year are:

1. Programming I
2. Programming II
3. Datastrukturer
4. Objektorienterad design

The courses are given sequentially from September to May

All of these courses, except objektorienterad design, can also be completed during the summer. Programming 1 and 2 can be completed through the FiTech network



# Programming 1

## Course Structure

Language: Python 3

Short so-called laboratory lectures (~45 min),  
hands-on exercises

Automatic evaluation of assignments in VILLE

Course exam in the end of the course

## Learning Outcomes

After completing the course, the students will be able to:

1. Analyze and explain the behavior of simple programs
2. Modify and further develop short programs that use various control structures
3. Choose appropriate conditional and loop structures for a given programming task
4. Construct algorithms that solve simple problems
5. Design, implement, test, and debug a program that uses the following fundamental programming constructs: basic arithmetic operations, simple I/O, conditional and loop statements, function definitions
6. Apply techniques of structured and modular programming to divide a program into smaller components
7. Develop programs in pair work

# Programming 1

## Assignments and Grading

Assignments done partially individually and partially in pairs

4 bigger assignments, done by pair programming.  
Manual grading by teaching assistants

Students can get up to grade 2 based on the points from the assignments

A programming exam in the end of the course can provide up to 3 additional grade levels

## Course Contents

Basic arithmetic operations

Data types and variables

I/O (keyboard/screen, later text files)

Functions (built-in, user-defined, calling functions)

Conditional statements

Loops

Lists

Text processing (strings)

Tuples, dictionaries (very briefly)

Text files

# Programming 2

## Course Structure

Language: Java

Focus is on object orientation

Approx. 100 course participants yearly

No teaching assistants – CodeGrade used for automatic evaluation of assignments

8 lectures + 16 supervised programming sessions

Assignments + exam

## Learning Outcomes

After completing the course, the students will be able to:

1. Create their own classes and objects
2. Create simple object-oriented programs that use class inheritance, encapsulation, and polymorphism
3. Structure their object-oriented programs using classes and packages
4. Create programs using abstract classes and Java interfaces
5. Make use of the Java API to incorporate ready-made functionality in their programs

# Programming 2

## Assignments and Grading

Assignment 1: 20 methods to implement. Done individually

Assignments 2-4: Create a small program in each assignment consisting of a few classes. Students are given some ready code for these assignments. Done in pairs

Exam: theoretical exam, i.e., no programming

Up to grade 3 is obtained from assignments, but only if students pass the exam.

The exam can increase the grade by 2 levels

## Course Contents

Basics of Java – Syntax, classes and objects, methods, variables

Object state and behavior

Encapsulation

Java API

Inheritance and polymorphism

Abstract classes and methods, interfaces

The stack and the heap

Constructors, life and death of objects

Static and final keywords

Exceptions

Java IO

# Datastrukturer

## Course Structure

Language: Java

Approx. 80 course participants yearly

No teaching assistants – CodeGrade, VILLE and moodle quizzes used for evaluating course assignments

Approx. 14 lecture + 6 supervised exercise sessions

assignments + exam

## Learning Outcomes

1. Define and explain the concept of time complexity, and apply simple complexity analysis
2. Define and explain an abstract data type (ADT)
3. Implement generic algorithms and data structures in Java
4. Define, explain, and implement data structures
5. Assess the advantages and disadvantages of data structures
6. Effectively apply data structures to practical programming problems
7. Implement basic sorting algorithms and describe their operating principles as well as their time and space complexity

# Datastrukturer

## Assignments and Grading

All 6 assignments are a combination of theoretical and practical tasks – Programming done in each of them

All assignments are done individually

Up to grade 2 is obtained from assignments, but only if students pass the exam.

The exam can increase the grade by 3 levels

## Course contents

Big-O notation

Abstract data types

Stacks

Queues

Lists

Binary search trees and balanced trees

Hashing

Priority queues

Select, bubble and insertion sort

Heapsort

Mergesort

Quicksort

# Objektorienterad design

## Course Structure

Language: Java

Focus is on code structure, design principles and OO-design

Approx. 60 course participants yearly

Approx. 14 lectures + 8 supervised programming sessions

2 teaching assistants – Help with grading assignments and giving written feedback

No exam

## Learning Outcomes

Domain modelling and testing

UML class diagrams

Design by Contract

Design principles

GUI-programming with JavaFX

UML state diagrams

Design patterns

Model-View-Controller

Strategy, Factory Method, Abstract Factory,

Observer, Template Method, Command,

Memento, Singleton, Decorator, and State Patterns

# Objektorienterad design

## Assignments and Grading

Students create a larger desktop application with a GUI. Split into 4 separate assignments. Done in pairs

Students get written feedback on their solutions regarding code quality, design, and functionality

Giving the feedback is very time consuming

AI cannot solve the assignments in this course, at least not yet...

## Course Contents

Domain modelling and testing

UML class diagrams

Design by Contract

Design principles

GUI-programming with JavaFX

UML state diagrams

Design patterns

Model-View-Controller

Strategy, Factory Method, Abstract Factory,

Observer, Template Method, Command,

Memento, Singleton, Decorator, and State Patterns

# Artificial Intelligence in the Courses

Using AI for completing the assignments is not allowed in any of these courses

Students are allowed to discuss programming concepts with AI

AI is currently not used as part of the programming education in any way



# Other Programming Courses @ ÅA

## Programvaruproduktion

For 2<sup>nd</sup> and 3<sup>rd</sup> year students

Students create a larger SW system in groups of 4 persons

Work according to Scrum. Little guidance by the lecturer. Lecturer plays the role of the customer

Students are allowed to use AI for any purposes in the course

## Programmering med AI (planned)

Developing a new advanced course in programming

Will be the last programming course students take

Currently planned contents include:

1. Generating, refactoring and documenting code with AI
2. AI for testing and debugging
3. Project work while assisted by AI

**Thank you for your attention!**



