UNIVERSITY OF GOTHENBURG

# DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

## DIT143 Functional Programming, 7.5 credits

Funktionell programmering, 7,5 högskolepoäng

*First Cycle*

---

### Confirmation

This course syllabus was confirmed by Department of Computer Science and Engineering on 2017-12-19 and was last revised on 2018-03-26 to be valid from 2018-08-19, autumn semester of 2018.

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

### Position in the educational system

The course is offered within several programmes. It is also a single subject course at the University of Gothenburg.

The course can be part of the following programmes: 1) Mathematical Sciences, Master's Programme (N2MAT), 2) Computer Science, Master's Programme (N2COS), 3) Applied Data Science Master's Programme (N2ADS), 4) Bachelor's Programme in Mathematics (N1MAT) and 5) Software Engineering and Management Master's Programme (N2SOF)

| *Main field of studies* | *Specialization* |
|---|---|
| Computer Science | G1F, First Cycle, has less than 60 credits in first-cycle course/s as entry requirements |

### Entry requirements

To be eligible for this course, students must have successfully completed a 7.5 credits course in programming in a paradigm other than functional, e.g., DIT948 Programming, DIT042 Object-Oriented Programming, MVG300 Programming in Matlab, 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

After completion of the course the student is expected to be able to:

*Knowledge and understanding*

- describe the basic concepts of modern functional programming languages, such as: data types, first- and higher-order functions, lazy evaluation, infinite data structures
- describe a basic repertoire of functional programming techniques, such as: recursion, testing, the role of data types in modelling and problem solving

*Competence and skills*

- write small to medium-sized functional programs for a variety of applications
- exploit a variety of programming techniques typical in functional programming, such as: use of recursion, modelling with recursive datatypes, abstraction and reuse with the help of higher order functions, polymorphism and monads
- implement effective tests for functional programs with help of suitable tools

*Judgement and approach*

- show the ability, in various contexts, to judge which programming techniques are most appropriate for solving the problem at hand
- identify the strengths and possible weaknesses of the functional programming paradigm, compared to other programming paradigms

## Course content

This course introduces a functional programming language and related concepts to students that already have some knowledge in programming. The primary goal is to make it possible for students to apply the functional approach to write realistic (small- to medium-sized) programs, and transfer general programming skills to the functional programming domain.

Concrete topics encountered in the course include:

- functions as first-class values
- compound data types (lists, tuples, and user-defined types)
- recursion and recursive data types
- using abstraction to avoid repetitive programming
- algebraic data types
- polymorphism and type classes

- time complexity: efficient and inefficient functional programs
- pure functions vs input-output
- modules and abstract data types
- testing functional programs
- lazy evaluation and infinite objects
- monads

*Sub-courses*

1. **Written examination** *(Skriftlig tentamen),* 4.5 higher education credits
   Grading scale: Pass with Distinction (VG), Pass (G) and Fail (U)
2. **Laboratory work** *(Laboration),* 3 higher education credits
   Grading scale: Pass (G) and Fail (U)

### Form of teaching

Teaching is through lectures and self-study exercises, supported by drop-in consultation times and on-line help.

*Language of instruction:* English

### Assessment

The course is examined by an individual written exam carried out in an examination hall at the end of course, and a number of compulsory assignments typically 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).
A Pass grade (G) for the entire course requires at least a Pass grade for all sub-courses.

To be awarded Pass with Distinction (VG) for a full course, the student must, in addition, receive the grade VG on the sub-course Written examination.

**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**

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 DIT142, 7.5 credits. The course cannot be included in a degree which contains DIT142. Neither can the course be included in a degree which is based on another degree in which the course DIT142 is included.

The course cannot be included in a degree programme where course DIT440 Introduction to Functional Programming is included, or any other degree which is based on a degree which includes DIT440.