



## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

### **DIT083 Testing, Debugging and Verification, 7.5 credits**

Testning, felsökning & verifiering, 7,5 högskolepoäng

*First Cycle*

---

#### **Confirmation**

This course syllabus was confirmed by Department of Computer Science and Engineering on 2019-11-25 to be valid from 2020-08-31, autumn semester of 2020.

*Field of education:* Science 100%

*Department:* Department of Computer Science and Engineering

#### **Position in the educational system**

The course Testing, Debugging and Verification, 7.5 higher education credits, is a part of the Computer Science Bachelor's programme. It is also a single subject course at the University of Gothenburg.

The course can be part of the following programmes: 1) Computer Science, Master's Programme (N2COS), 2) Applied Data Science Master's Programme (N2ADS), 3) Computer Science, Bachelor's Programme (N1COS) and 4) Software Engineering and Management Master's Programme (N2SOF)

*Main field of studies*

Computer Science

*Specialization*

G1F, First cycle, has less than 60 credits in first-cycle course/s as entry requirements

#### **Entry requirements**

To be eligible for the course students should have successfully completed 45 hec of an education aiming at a bachelor degree within Computer Science, Software Engineering or equivalent.

Within these 45 hec, the student should have successfully completed:

- A 7.5 hec course in discrete mathematics (such as DIT980)
- A 7.5 hec course in imperative/object-oriented programming (such as DIT012 or DIT953)

## Learning outcomes

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

### *Knowledge and understanding*

- Describe the possibilities and limitations of both informal and formal techniques for the discovery, analysis, and resolving of program errors.
- Describe the concept of precise specifications of software units.
- Discuss the principles of software verification.

### *Competence and skills*

- Express precise specifications of software units.
- Systematically discover inputs on which a program fails.
- Locate, analyse, and fix the error which caused the failure.
- Formally reason about simple programs for guaranteeing correctness.

### *Judgement and approach*

- Judge the various error removal and prevention practices that one can encounter in a working context.
- Contribute to the development and improvement of error removal and prevention techniques in a given context.
- Follow up on, and take advantage of, conceptual and technical developments in the area of testing, debugging, and verification that go beyond the exact techniques covered in the course.

## Course content

The main aim of the course is to provide a basic understanding of techniques that cope with errors in programs. Recurring themes are a) the identification of errors, b) their analysis, and c) their removal. The course also provides an understanding of systematic ways to convince oneself that a program unit really does what it should.

The course covers formal and informal methods, testing (terminology, coverage, unit tests, a unit test framework), debugging (control, workflow, localisation, tools), formal specifications (pre-/postconditions, invariants), formal verification (logics, tool support). Throughout, the course is concerned with imperative programs in general, and object-oriented programs in particular.

After the course, student have understood - and are able to employ - the methods testing (trying to reveal the presence of errors in a systematic way), debugging (the act of isolating and fixing errors), and verification (reasoning about programs in order to guarantee correctness). All these methods only make sense in the presence of a

specification of what the program is supposed to do.

#### *Sub-courses*

1. **Written hall examination** (*Skriftlig salstentamen*), 5 credits  
Grading scale: Pass with Distinction (VG), Pass (G) and Fail (U)
2. **Laboratory work** (*Laboration*), 2.5 credits  
Grading scale: Pass (G) and Fail (U)

#### **Form of teaching**

The course consists of lectures, exercise classes and laboratory work.

*Language of instruction:* English

#### **Assessment**

The course is examined by a final individual written hall examination 5.0 hec at the end of the course, and laboratory work (hand-ins) 2.5 hec. The laboratory work (hand-ins) is usually done in groups of 2-3 students.

In case a hand-in is not passed in the first attempt, the student is granted a second try.

*A student who has failed two examinations on the same material has the right to request a change of examiner. Such a request must be submitted to the Department in writing and shall be granted unless there are particular reasons not to do so.*

*In cases where a course has been discontinued or has undergone major changes, students must be guaranteed at least three examination opportunities (including the regular opportunity) based on the previous content of the course for a period of at least one year.*

#### **Grades**

The grading scale comprises: Pass with Distinction (VG), Pass (G) and Fail (U).

The grading scale comprises: Pass with Distinction (VG), Pass (G) and Fail (U).

In order to get the grade Pass for the full course the student must pass both the Written Exam and the laboratory work (hand-ins).

To be awarded the grade Pass with Distinction the student must get the grade Pass with Distinction on the written exam and pass the laboratory work (hand-ins).

#### **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 will be announced at the latest 8 weeks prior to the start of the course.

The course replaces the course DIT082 Testing, debugging and verification, 7.5credits.

The course cannot be included in a degree which contains DIT082. Neither can the course be included in a degree which is based on another degree in which the course DIT082 is included.

The course cannot be included in a degree which contains DIT635. Neither can the course be included in a degree which is based on another degree in which the course DIT635 is included