



## COMPUTER SCIENCE AND ENGINEERING

### **DIT082 Testing, Debugging and Verification, 7.5 credits**

Testing, Debugging and Verification, 7,5 högskolepoäng

*First Cycle*

---

#### **Confirmation**

This course syllabus was confirmed by The IT Faculty Board on 2007-10-23 and was last revised on 2017-06-07 by Department of Computer Science and Engineering to be valid from 2017-08-20, autumn semester of 2017.

*Field of education:* Science 100%

*Department:* 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 and an elective 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) Software Engineering Master's Programme (N2SOM), 4) Computer Science, Bachelor's Programme (N1COS) and 5) No translation available (NDATM)

*Main field of studies*

Computer Science

*Specialization*

G2F, First Cycle, has at least 60 credits in first-cycle course/s as entry requirements

#### **Entry requirements**

To be eligible for the course students should have successfully completed:

- one year of an education aiming at a bachelor degree within Computer Science or equivalent
- a course in discrete mathematics (such as MMGD10)

- a course in object oriented programming (such as DIT011 or DIT950)

### **Learning outcomes**

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

#### *1. 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 verification.

#### *2. Skills and abilities*

- 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

#### *3. Judgement and approach*

- judge the various error removal and prevention practices they encounter in a work 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 for techniques that cope with errors in programs. Reoccurring 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 processing contrasting formal and informal methods, testing (terminology, coverage, unit tests, a unit test framework), debugging (control, workflow, localisation, tools), formal specifications (pre-/postconditions, invariants), automatic test case generation, formal verification (logics, tool support). Throughout, the course is concerned with imperative programs in general, and object-oriented programs in particular.

After the course, students 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.

### **Form of teaching**

*Language of instruction:* English

The course is given in English.

### **Assessment**

The course is examined by a final individual written hall exam, 5.0 hec, and Laboration (practical hand-in assignments), 2.5 hec. The laborations are usually done in groups of 2-3 students.

In case a practical hand-in assignment 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).

In order to get the grade Pass for the full course the student must pass both the Written Exam and laboration (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 Laboration (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.