

Datalogi - Fagmodulkursus 4 - Essential Computing II

Om kurset

uddannelse	Fagmodul i Datalogi
Kurstype	fagmodulkursus
Undervisningssprog	English
	Register through STADS Self-Service
	Registration will take place during the period October 28 - November 15, 2015
Tilmelding	Questions regarding registration please contact course secretary Anja Kastrup Christiansen (anjakc@ruc.dk)
	The course description is preliminary
Kursus starter	11-02-2016
Kursus slutter	21-04-2016
Undervisningstidspunkt	Thursdays 8.30-12.45 during weeks 6-16 (no lecture on March 24, 2016 (week 15) due to Easter Holiday)
Undervisningssted	43.2-029
forudsætninger	Students should have knowledge of either an object-oriented or procedural language. Knowledge of basic programming language features, including primitive data types, operators, control structures, functions (methods), and input/output is assumed Students are expected to be able to read texts in English at a level at least equivalent to High School B-level
Kursusrækkefølge	It is recommended that the subject module course 1 'Essential Computing I' is followed before attending this course
formål	The purpose of this course is to provide a practical introduction to algorithms and data structures from the viewpoint of abstract thinking and problem solving. The primary focus is on problem-solving techniques that allow the construction of sophisticated time-efficient programs.
Indhold	Teori og praksis for fundamentale algoritmer og datastrukturer; beregnelighed; kompleksitet og skalerbarhed af programmer; faser i IT systemudvikling fra specifikation til afprøvning.

- Hvordan et systems funktionalitet deles op med modulært design (How to apply modular design splitting up the system functionality)
- Hvorfor abstraktion er et nøgleprincip i design af genbrugelig og “evolvable” software (Why abstraction is a key principle in designing reusable, evolvable software)
- Fundamentale algoritmer og datastrukturer for effektiv lagring og fremfindelse af information (Fundamental algorithms and data structures for efficient information storage and retrieval)
- Avancerede kontrolstrukturer. Rekursion. Håndtering af undtagelser. Sempel programmering med tråde (Advanced control constructs. Recursion. Exception handling. Simple thread programming)
- Principper for opbygning af skalerbare systemer. Introduktion til beregnelighed og kompleksitet (The principles of building systems that scale up. Introduction to computability and complexity)
- Hvordan afprøvning designes og udføres (How to design tests and carry them out)
- Hvordan et system integreres og pakkes mhp. distribution og installation (How to integrate the system and package it for distribution and installation)

Undervisningsform The course is a theoretical course with a combination of lectures and problem solving

The goal of the course is for the students to acquire:

Knowledge:

- Knowledge about theory and practice for fundamental algorithms and data structures
- Knowledge about algorithm and problem complexity
- Knowledge about program development from specification to testing

Skills:

- Skills to program in an object-oriented programming language by means of techniques for modularization and abstraction
- Skills to systematically test software

Competencies:

- Competence to design and develop component-based software that is robust and scalable
- Competence to choose between different solutions and argue for the choice

To be more specific, on successful completion of the course, the student should be able to:

- Design, implement and test small to medium sized applications in an object-oriented programming language

bedømmelseskriterier

- Understand and clearly explain the mechanics of algorithms and data structures involving manipulation of references, nested loops and recursion
- Choose among and make use of the most important algorithms and data structures in libraries
- Explain how fundamental algorithms for data structures for searching and sorting may be implemented
- Analyze time and space usage of algorithms and data structures
- Reason about the correctness of an algorithm
- Apply the following algorithmic techniques when solving a problem: Divide-and-conquer, dynamic programming, backtracking

Individual oral exam with a duration of 15 minutes based on an individual written mini project (programming assignments) which must be handed in during the semester. The programming assignments are based on a given problem.

Eksamensform

The grading is a total of the mini project and the oral exam.

Examiners: Internal

Assessment: The 7-step marking scale

Same as the ordinary exam

Written assignment:

Wednesday at 12.00, June 29 2016 at OnlineEksamen.ruc.dk

Reksamensform

Oral exam:

August 23 (Tuesday), 2016

You will be informed of details such as exact time and location later

Written assignment:

Monday at 12.00, April 25 2016 at OnlineEksamen.ruc.dk

Eksamenstidspunkt

Oral exam:

June 6 or June 7, 2016

You will be informed of details such as exact time and location later

Aktivitetsansvarlig

Mads Rosendahl (madsr@ruc.dk)

Kursussekretær

Heidi Lundquist (heilu@ruc.dk)

Anja Kastrup Christiansen (anjakc@ruc.dk)

Underviser Ebbe Vang (ebbevang@ruc.dk)

STADS fagmodulskursus

stamdata belastning : 5 ECTS

aktivitetskode : U25234

prøveform : Skriftlig/mundtlig

bedømmelse : 7-trinsskala

censur : Intern censur