Computer Science

Find another programme

First year courses

BSc Computer Science year 1 core courses

Dept. of Advanced Computing Sciences

Introduction to Computer Science

Full course description

The primary goal of Introduction to Computer Science is to introduce fundamental concepts and foster critical skills found throughout the field of computer science. Fundamental concepts include algorithms, computer architecture and hardware, models of computation, computer networks, and operating systems. Critical skills include abstraction, decomposition, pattern recognition, and algorithmic thinking. All concepts and skills are introduced in a lecture setting and explored further in the lab through the development of a wirelessly controlled microcontroller device. At the end of this course, students will appreciate the depth of the field and be prepared for subsequent research and educational activities.

Prerequisites

None.

Recommended reading

- "Computational Thinking for the Modern Problem Solver" by David Riley, Kenny A. Hunt
- "Computer Science Illuminated" by Nell B. Dale

BCS1110
Period 1
4 Sep 2023
27 Oct 2023
Print course description
ECTS credits:
4.0
Coordinators:

- A.R.K. Sai
- T.A. Bitterman

Teaching methods:
Lecture(s)
Assessment methods:
Written exam, Assignment
Dept. of Advanced Computing Sciences

Procedural Programming

Full course description

The course provides the basics of computer science and computer programming. After a short introduction to computer organization, the principles of programming are presented. The main topics of the course are: data types, variables, methods, parameters, decision structures, iteration, arrays, recursion and a branching application (related to the semester project). Programming skills will be acquired during practical sessions using the object-oriented programming language Java.

Prerequisites

None. It appears as part of the pre-requisites of the second semester project in year 1, both projects of year 2, the year 2 course Databases and the year 3 courses, Parallel Programming and Robotics and Embedded Systems.

The course appears as desired prior knowledge for the courses Introduction to Objects in Programming, Data Structures and Algorithms, Software Engineering, Databases and Machine Learning.

Recommended reading

H. Schildt, Java: A Beginner's Guide, Eighth Edition, ISBN: 1260440214, McGraw-Hill Education

BCS1120
Period 1
4 Sep 2023
27 Oct 2023
Print course description
ECTS credits:
4.0
Coordinators:

- E. Hortal Quesada
- T.A. Bitterman
- D.H. Cámpora Pérez

Teaching methods:
Project-Centered Learning
Assessment methods:
Written exam, Assignment
Dept. of Advanced Computing Sciences

Discrete Mathematics

Full course description

In this course, we build a mathematical framework that is based on logic and reason. The main objective of the course is to make students familiar with the language of mathematics. Students will

learn how to make sound arguments and to detect where and why certain arguments go wrong. For this purpose, we will discuss the basic principles of logic and, closely related, the basic types of mathematical proofs. In doing so, we will encounter numbers such as integers, natural numbers and real numbers and we shall examine what makes these numbers special. After that, we will use basic logic to discuss, among other things, the following mathematical concepts: infinity, sets, relations, functions, permutations and combinations. Our fundamental tool in all of this is plain common sense. You really do not need your toolbox of mathematical formulas learned in previous studies and neither do you need a calculator. Pen and paper are the basic instruments needed. After completing each topic, exercises will be provided to be completed in class or at home, since mathematics is mainly learned by practising repeatedly.

Prerequisites

None.

Recommended reading

None.

BCS1130
Period 1
4 Sep 2023
27 Oct 2023
Print course description
ECTS credits:
4.0
Coordinators:

- M. Musegaas
- O. D'Huys

Teaching methods:
Project-Centered Learning
Assessment methods:
Written exam
Dept. of Advanced Computing Sciences

Objects in Programming

Full course description

This course is a follow-up of the course Procedural Programming. It teaches object-oriented programming in Java. The main topics covered in the course are objects and classes, interfaces and polymorphism, event handling, inheritance, graphic user interfaces, exception handling, and streams.

Computer Science

Prerequisites

Desired prior knowledge: Basic Java Programming

Recommended reading

C. Horstmann (2016). Java Concepts (8th Edition). John Wiley & Sons, New York, ISBN: 978-1-1190-5645-4 C. Horstmann (2012). Big Java Late Objects. John Wiley & Sons, New York, ISBN 978-1-1180-8788-6

BCS1220
Period 2
30 Oct 2023
22 Dec 2023
Print course description
ECTS credits:
4.0
Coordinators:

- T.A. Bitterman
- E.N. Smirnov
- F. Barile

Teaching methods:
Project-Centered Learning
Assessment methods:
Written exam, Assignment
Dept. of Advanced Computing Sciences

Calculus

Full course description

The following subjects will be discussed in Calculus: limits and continuity, differential calculus, inverse and transcendental functions, mean value theorem, integral calculus, sequences and series, introduction to differential equations, introduction to multivariable calculus. In addition to the main facts and concepts, problem-solving strategies will be discussed. Both the intuition behind the concepts and their rigorous definitions will be presented along with simple examples of formal mathematical proofs.

Prerequisites

None

Recommended reading

None

BCS1440

Computer Science

Period 2
30 Oct 2023
22 Dec 2023
Print course description
ECTS credits:
4.0
Coordinators:

- O. D'Huys
- G.M. Schoenmakers
- M. Boussé

Teaching methods:
Project-Centered Learning
Assessment methods:
Written exam
Dept. of Advanced Computing Sciences

Logic

Full course description

This course deals with three logical systems, namely propositional logic, first-order predicate logic and epistemic logic. The course covers notation systems, syntax and semantics, valid consequences, deduction, semantic tableaux, and proof systems.

Prerequisites

None

Recommended reading

None

BCS1530
Period 2
30 Oct 2023
22 Dec 2023
Print course description
ECTS credits:
4.0
Coordinators:

- T.D. Rienstra
- S.J. Maubach
- N. Roos

Teaching methods:
Project-Centered Learning
Assessment methods:

Computer Science

Written exam, Assignment
Dept. of Advanced Computing Sciences

Project 1-1

Full course description

Students work on a project assignment in small groups of about six students. The group composition stays the same for the whole project and is announced shortly before the project opening in period 1.1. The students are guided through the project by a fixed tutor. The project assignment is divided into three subtasks (one per period) and is strongly related to the content of the courses from period 1.1 and 1.2. In period 1.1, after receiving the assignment for the whole project at the end of week 5, the students work full-time on the project in week 6. In this week, each group meets the tutor twice. In period 1.2, the students continue working on the project, while also having to attend the courses of that period. They meet their tutor approximately once a week. In period 1.3, the students work three weeks full-time on the project and meet their tutor about twice a week.

At the beginning of period 1.2 and 1.3, the students have to hand in a planning for the current phase. At the end of each period, the students have to give a presentation and the source code, presentation and an overview of who did what need to be uploaded to Canvas. While the presentations at the end of period 1.1 and 1.2 are in front of the examiners and the tutors, the presentations at the end of period 1.3 will additionally be in front of the fellow students. In period 1.3, they furthermore have to hand in a report and attend a product and report examination.

Project 1-1 will start in period 1.1 and period 1.2. The credits for the projects will become available at the end of period 1.3.
For each period, we will give a short explanation of the various parts. Before the start of each period, the students will receive detailed information about the content, the study material, the teaching form, the schedule, and the examination method.

Prerequisites

This project has no prerequisites. This project occurs as part of the prerequisites of project 2-1.

Recommended reading

None

BCS1300
Semester 1
4 Sep 2023
2 Feb 2024
Print course description
ECTS credits:
6.0
Coordinator:

- M. Boussé

Teaching methods:

Computer Science

Project-Centered Learning, Work in subgroups, Presentation(s), Skills
Assessment methods:
Assignment, Presentation and paper, Participation
Dept. of Advanced Computing Sciences

Linear Algebra

Full course description

This course introduces the fundamental concepts of linear algebra, and examines them from both an algebraic and a geometric point of view. First, we address what can be recognized without doubt as the most frequently occurring mathematical problem in practical applications: how to solve a system of linear equations. Then we discuss linear functions and mappings, which can be studied naturally from a geometric point of view. Vectors spaces are then introduced as a common framework that brings all themes together. Next, we shift from the geometric point of view to the dynamic perspective, where the focus is on the effects of iterations (i.e., the repeated application of a linear mapping). This involves a basic theory of eigenvalues and eigenvectors, which have many applications in various branches of science as for instance in problems involving dynamics and stability, in control theory, and in optimization problems found in data science. Key concepts in the course are vectors, matrices, systems of linear equations, eigenvalues, eigenvectors, linear transformations, and orthogonality. The software package Matlab is introduced in the accompanying computer classes, where emphasis is put on the application of linear algebra to solve real world problems.

Prerequisites

None.

Recommended reading

None.

BCS1410
Period 4
5 Feb 2024
5 Apr 2024
Print course description
ECTS credits:
4.0
Coordinators:

- M. Musegaas
- P.W.L. Dreesen
- S.A. Chaplick

Teaching methods:
Project-Centered Learning
Assessment methods:
Written exam
Dept. of Advanced Computing Sciences

Computer Science

Data Structures and Algorithms

Full course description

As a continuation of the courses Procedural Programming and Objects in Programming, this course will treat the systematic design and application of data structures and algorithms.
Data structures such as lists, trees, graphs, and dictionaries, the associated algorithms and their complexity are explored in this course. Algorithms for applications such as sorting, pattern matching and graph
traversal are also part of the course. Furthermore, design principles for algorithms such as recursion, divide-and-conquer and dynamic programming will be treated as well. Furthermore, students will develop skills to analyse the run-time and space complexity of data structures and algorithms.

Prerequisites

None.

Desired Prior Knowledge: Discrete Mathematics, Procedural Programming and Objects in Programming.

Recommended reading

Required Reading: Sedgewick and Wayne (2011) Algorithms Fourth Edition. Addison Wesley. ISBN: 978-0321573513

Recommended Reading: A Y Bhargava (2016). Grokking Algorithms: An Illustrated Guide for Programmers and Other Curious People. Manning. ISBN: 978-1617292231

BCS1420
Period 4
5 Feb 2024
5 Apr 2024
Print course description
ECTS credits:
4.0
Coordinators:

- T.H.J. Pepels
- F. Barile
- D.H. Cámpora Pérez

Teaching methods:
Project-Centered Learning
Assessment methods:
Written exam, Assignment
Dept. of Advanced Computing Sciences

Object-Oriented Modelling

Full course description

This course introduces students to the design and analysis aspects of object-oriented programming. Software construction for real world applications has inherent complexities both in terms of designing and maintaining it. In this course, the students will learn how to model a real-world problems in an object-oriented programming context using tools like Unified Modelling Language (UML). Students will also learn techniques such as structural, behavioral and creational design patterns, GRASP principles to create modular, flexible and reusable software. After completing the course, the students would have gained practical experience in problem formulation, decomposition (analysis) and solution building (design) using object-oriented modelling techniques.

Prerequisites

None.

Recommended reading

- "Clean Code: A Handbook of Agile Software Craftsmanship" by Robert Cecil Martin
- "Design Patterns: Elements of Reusable Object-Oriented Software" by Erich Gamma et al.

BCS1430
Period 4
5 Feb 2024
5 Apr 2024
Print course description
ECTS credits:
4.0
Coordinators:

- A.R.K. Sai
- Y. Wang

Teaching methods:
Lecture(s)
Assessment methods:
Written exam, Assignment
Dept. of Advanced Computing Sciences

Databases

Full course description

This course introduces students to fundamental concepts of databases with a focus on relational database systems. Students will learn how to design, implement and optimize a relational database. The course provides a comprehensive study of data modeling, data description languages, and query facilities such as relational algebra and SQL. Students will also be exposed to the internal workings of database management systems with an overview of topics such as concurrency, recovery,

indexing, and triggers. The course will cover current topics related to managing Big Data using alternate data models such as NoSQL. The students will apply their knowledge on these topics in designing and implementing a database system as a part of a group project.

Prerequisites

None.

Recommended reading

"Readings in Database Systems" by Peter Bailis, Joseph M. Hellerstein, Michael Stonebraker, 5th Edition.

BCS1510
Period 5
8 Apr 2024
7 Jun 2024
Print course description
ECTS credits:
4.0
Coordinator:

- A.R.K. Sai

Teaching methods:
Lecture(s)
Assessment methods:
Written exam, Assignment
Dept. of Advanced Computing Sciences

Statistics

Full course description

: Statistics introduces the student to the main concepts of both probability theory and statistics. With respect to probability theory, students learn how to make use of random variables to extract the probability distribution of an experiment. Additionally, topics such as expectation, standard deviation, and independence will be discussed. The statistics part of the course discusses basic statistical topics such as the central limit theorem, verification of hypotheses, and confidence intervals. After completing this course students will have obtained an overview of commonly seen probability distributions, as well as several statistical procedures. Additionally, the student will be able to deal with problems that involve probabilities and determine an outcome for such problems (e.g., the expected outcome).

Prerequisites

None.

BCS1520

Computer Science

Period 5
8 Apr 2024
7 Jun 2024
Print course description
ECTS credits:
4.0
Coordinator:

- M.C. ten Thij

Teaching methods:
Lecture(s)
Assessment methods:
Written exam, Take home exam
Dept. of Advanced Computing Sciences

Algorithmic Design

Full course description

Algorithmic Design formalizes the main algorithmic paradigms and techniques including greedy and divide-and-conquer strategies, dynamic programming, multi-dimensional searching, computational geometry, linear programming, randomization, and approximation algorithms. It familiarizes students with amortization and NP-completeness. After completing the course, students will be expected to show good design principles and adequate skills at reasoning about the correctness and complexity of algorithms.

Prerequisites

None.

Recommended reading

Goodrich and Tamassia (2015) Algorithm Design and Applications. Wiley. ISBN: 978—1-118-33591-8

BCS1540
Period 5
8 Apr 2024
7 Jun 2024
Print course description
ECTS credits:
4.0
Coordinators:

- T.A. Bitterman
- Y. Wang

Teaching methods:
Lecture(s)
Assessment methods:

Computer Science

Written exam, Assignment
Dept. of Advanced Computing Sciences

Project 1-2

Full course description

Students work on a project assignment in small groups of about six students. The group composition stays the same for the whole project and is announced before the project opening in period 1.4. The students are guided through the project by a fixed tutor. The project assignment is divided into three subtasks (one per period) and is strongly related to the content of the courses from period 1.4 and 1.5. In period 1.4, after receiving the assignment for the whole project at the end of week 5, the students work full-time on the project in week 6. In this week, each group meets the tutor twice. In period 1.5, the students continue working on the project, while also having to attend the courses of that period. They meet their tutor approximately once a week. In period 1.6, the students work three weeks full-time on the project and meet their tutor twice a week.

At the beginning of period 1.5 and 1.6, the students have to hand in a planning for the current phase. At the end of each period, the students have to give a presentation and the source code, presentation and an overview of who did what need to be uploaded to Canvas. While the presentations at the end of period 1.4 and 1.5 are in front of the examiners and the tutors, the presentations at the end of period 1.6 will additionally be in front of the fellow students. In period 1.6, they furthermore have to hand in a report and attend a product and report examination.

Project 1-2 will start in period 1.4 and period 1.5. The credits for the projects will become available at the end of period 1.6.
For each period, we will give a short explanation of the various parts. Before the start of each period, the students will receive detailed information about the content, the study material, the teaching form, the schedule, and the examination method.

Prerequisites

In order to participate in this project the student has to have passed two out of four courses from the set: Discrete Mathematics, Calculus, Procedural Programming and Objects in Programming.

This project occurs as part of the prerequisites of project 2-2.

Recommended reading

None.

BCS1600
Semester 2
5 Feb 2024
5 Jul 2024
Print course description
ECTS credits:
6.0
Coordinator:

Computer Science

- O. D'Huys

Teaching methods:
Project-Centered Learning, Work in subgroups, Presentation(s), Skills
Assessment methods:
Assignment, Presentation and paper, Participation