Course Outlines
Year 1

ALGORITMIEK
This course treats various methods to design and analyze datastructures and algorithms for a wide range of problems. The most important new datastructure treated is the graph, and the general methods introduced are: greedy algorithms, divide and conquer, dynamic programming and network flow algorithms. These general methods are explained by a number of concrete examples, such as simple scheduling algorithms, Dijkstra, Ford-Fulkerson, minimum spanning tree, closest-pair-of-points, knapsack, and Bellman-Ford. Throughout this course there is significant attention to proving the correctness of the discussed algorithms. All material for this course is in English. The recorded lectures, however, are in Dutch.
Subjects
01. Introduction Algoritmiek
02. Graphs
03. Greedy algorithms
04. Divide and conquer
05. Dynamic programming
06. Network flow
Readings
Activities
Exams

AUTOMATED SOFTWARE TESTING: ADVANCED SKILLS FOR JAVA DEVELOPERS
Software testing gets a bad rap for being difficult, time-consuming, redundant, and above all – boring. But in fact, it is a proven way to ensure that your software will work flawlessly and can meet release schedules.

In a two-course series, we will teach you automated software testing in an inspiring way. We will show you that testing is not as daunting a task as you might think, and how automated testing will make you a better developer who programs excellent software.

This second course builds upon the first course's material. It covers more advanced tools and techniques and their applications, now utilizing more than just JUnit. Key topics include Test-Driven Development, state-based and web testing, combinatorial testing, mutation testing, static analysis tools, and property-based testing.

This is a highly practical course. Throughout the lessons, you will test various programs by means of different techniques. By the end, you will be able to choose the best testing strategies for different projects..

If you are or want to become a five-star software developer, QA engineer, or software tester, join this course. Testing will never be the same again!

This course is part of the Professional Certificate Program Automated Software Testing in Java.
- The key ideas behind advanced software testing techniques, such as mutation and property-based testing
- The current state-of-the-art in software testing research, such as the usage of artificial intelligence to automate testing activities
- The limitations of current testing techniques, and how to determine the best testing strategies for a given context

AUTOMATED SOFTWARE TESTING: PRACTICAL SKILLS FOR JAVA DEVELOPERS

Software testing gets a bad rap for being difficult, time-consuming, redundant, and above all – boring. But in fact, it is a proven way to ensure that your software will work flawlessly and can meet release schedules.

In a two-course series, we will teach you automated software testing in an inspiring way. We will show you that testing is not as daunting a task as you might think, and how automated testing will make you a better developer who programs excellent software.

This first course will teach you how to master software testing and software quality assurance using JUnit. Key topics of the first course will include unit testing, test adequacy and code coverage, mock objects, design for testability, and test code quality.

This is a highly practical course. Throughout the lessons, you will test various programs by means of different techniques. By the end, you will be able to choose the best testing strategies for different projects.

If you are or want to become a five-star software developer, QA engineer, or software tester, join this course. Testing will never be the same again!

How to unit test any software system using current state-of-the-art tools
How to derive test cases that deal with exceptional, corner, and bad-weather cases by means of several different techniques
The limitations of current testing techniques and how to decide on the best testing strategies for a given context
How to develop testable architectures and to write maintainable test code

GLOBAL SOFTWARE ENGINEERING

Software engineering operates ever more frequently in globally distributed settings, in a practice that is known as Globally Distributed Software Engineering (GDSE). In this course, you will obtain a practical overview of the organization and operation of software engineering of this practice. As such, it is aimed at professionals in distributed software development teams, and executives setting up and leading such teams who would like to develop the required technical and organizational skills.

The course covers the subject in an accessible and practical manner. Through video lectures, group assignments and exercises, you will be familiarized with the advantages and disadvantages of GDSE, the practical consequences of GDSE and its technological feasibilities and infeasibilities. You will learn about real-world experiences of users and examples of GDSE applications such as outsourcing, offshore software development, near-shoring and multi-partner systems development.

You will apply the knowledge gained through hands-on experience with GDSE by working together with team members from different countries as a distributed team; and through analysis of best-practice examples. Together with other course participants you will prepare a number of artefacts that build on the body of knowledge of GDSE and so have the chance to contribute to this growing field of knowledge.

Guest lectures from industry experts and researchers will be an integral part of the course. These lectures will demonstrate how GDSE is handled in industry, how decision-makers lead their teams in this context, and what is the state-of-the-art in GDSE research

The course consists of seven main topics:

Globally Distributed Software Engineering
Distributed Agile
The Software Engineer and GDSE
Boardroom: C-level Decision-Making
GDSE Research
Time and Cultural Differences
GDSE Locations: Near-shoring and Offshoring

THE BUILDING BLOCKS OF A QUANTUM COMPUTER: PART 1
There is no doubt that the quantum computer and the quantum internet have many profound applications, they may change the way we think about information, and they could completely change our daily life.

But how do a quantum computer and a quantum internet work? What scientific principles are behind it? What kind of software and protocols do we need for that? How can we operate a quantum computer and a quantum internet? And which disciplines of science and engineering are needed to develop a fully working system?

In a series of two MOOCs, we will take you through all layers of a quantum computer and a quantum internet. The first course will provide you with the scientific basis by explaining the first layer: the qubits. We will discuss the four types of qubits that QuTech research center at Delft University of Technology focuses on: topological qubits, Spin qubits, Trans qubits and NV Centre qubits. We will teach you the working principles of qubits and, at the same time, the working principles of a computer made of these qubits.

In the upcoming second MOOC (yet to be announced), we will introduce the other layers needed to build a quantum computer and a quantum internet, such as the micro-architecture, compilers, quantum error correction, repeaters and quantum algorithms.

These two courses offer you an opportunity to deepen your knowledge by continuing the journey started in our first MOOC, which focused on the applications of a quantum computer and a quantum internet.

Note that these courses offer a full overview of the layers of a quantum computer and a quantum internet, and therefore they will not go into too much detail per layer. For learners seeking to fully understand one specific topic we can recommend other courses authored by QuTech:

In the field of Quantum Internet: Quantum Cryptography
In the field of topological phenomena: Topology in Condensed Matter: Tying Quantum Knots
This course is authored by experts from the QuTech research center at Delft University of Technology. In the center, scientists and engineers work together to enhance research and development in quantum technology. QuTech Academy's aim is to inspire, share and disseminate knowledge about the latest developments in quantum technology.

An overview of all building blocks of a quantum computer, from qubits and electronics all the way up to software.
The scientific principles behind the quantum computer that make quantum technologies possible.
How qubits can be used and controlled efficiently and the workings of the four most promising types of solid-state qubits: Superconducting Transmon Qubit, Silicon Spin Qubit, Diamond NV Center Qubit, and Topological Qubit.

THE BUILDING BLOCKS OF A QUANTUM COMPUTER: PART 2

There is no doubt that the quantum computer and the quantum internet have many profound applications, they may change the way we think about information, and they could completely change our daily life.

But how do a quantum computer and a quantum internet work? What scientific principles are behind it? What kind of software and protocols do we need for a quantum computer and a quantum internet? Which disciplines of science and engineering are needed to develop these? And how can we operate a fully working system?

In this series of two courses, we take you through all layers of a quantum computer and a quantum internet. In part 1 we explained the first layer: the qubits. We introduced the most promising quantum platforms and discussed how to do quantum operations on the physical qubits. In part 2 we will introduce the other layers needed to build and operate a quantum computer and a quantum internet, such as the quantum classical interface, micro-architecture, compilers, quantum error correction, networks and protocols and quantum algorithms.

These two courses offer you an opportunity to deepen your knowledge by continuing the journey started in our first course, which focused on the applications of a quantum computer and a quantum internet.

Note that these courses offer a full overview of the layers of a quantum computer and a quantum internet, and therefore they will not go into too much detail per layer. For learners seeking to fully understand one specific topic we can recommend other courses authored by QuTech:

In the field of Quantum Internet: Quantum Cryptography
In the field of topological phenomena: Topology in Condensed Matter
This course is authored by experts from the QuTech research center at Delft University of Technology. In the center, scientists and engineers work together to enhance research and development in quantum technology. QuTech Academy’s aim is to inspire, share and disseminate knowledge about the latest developments in quantum technology.

An overview of all building blocks of a quantum computer, including micro-architecture, compilers, quantum error correction, repeaters and quantum algorithms.
A deeper understanding of the building blocks of a quantum internet, and the protocols and networks needed to realize this.
The scientific principles behind the quantum computer that make quantum technologies possible.

THEORY OF COMPUTATION

Computability Theory deals with one of the most fundamental questions in computer science:

What is computing and what are the limits of what a computer can compute?

Or, formulated differently: “What kind of problems can be algorithmically solved?” During the course this question will be studied. Firstly, the notion of algorithm or computing will be made precise by using the mathematical model of a Turing machine. Secondly, it will be shown that basic issues in computer science, like “Given a program P does it halt for any input x?” or “Given two program P and Q, are they equivalent?” cannot be solved by any Turing machine. This shows that there exist problems that are impossible to solve with a computer, the so-called “undecidable problems”.

Subjects
1: Preliminaries – Automata and Languages
2: Turing Machines and the Church-Turing Thesis
3: Decidabillity
4: Reducibility
5: Self-reproduction
Readings Theory of Computation
Exams Theory of Computation