Open Universiteit

Detailed Course Descriptions

Erasmus+ course catalogue | Science - Computer Science

Erasmus+ course catalogue of the Open University of the Netherlands. Each course has an English title; the Dutch name, as used in our course offer, is shown as well. Descriptions have been translated into English.

Questions? Please contact subsidieadvies@ou.nl

Each course shows: type, language of the course, ECTS (study credits) and price in euros. A dash instead of a price means that no price is charged.

Science - Computer Science

120 courses and programmes in this document.

BTI-2026-2027 - Bachelor's programme in Computer Science (BSc)
CPPAI - CPP Certified Software Architect
CPPITSE - IT security engineer
FOICDI-2026-2027 - Focus programme Cybersecurity and Digital Innovation
IB0102 - Introduction to Computer Science
IB0203 - Premaster programming techniques 1: relational databases and object-oriented programming
IB0302 - Relational databases
IB0313 - Premaster programming techniques 2: object-oriented analysis, design and programming
IB0402 - Logic, sets and relations
IB0602 - Linear algebra and stochastics
IB0702 - Computer networks
IB0802 - Formal languages and automata
IB0902 - Advanced object-oriented programming
IB1002 - Object-oriented analysis and design
IB1102 - Object-oriented programming
IB1202 - Practicum design and implementation
IB1302 - Operating systems
IB1402 - Communication skills
IB1502 - Data structures and algorithms
IB1602 - Functional programming
IB1812 - Security and IT
IB2012 - Academic writing skills
IB2302 - Distributed algorithms
IB2602 - AppLab
IB2702 - Concepts of programming languages
IB2812 - Artificial intelligence
IB2912 - Logic and computer science
IB3112 - Software engineering
IB3202 - Software testing
IB3502 - Premaster AI: programming with Python
IB3702 - Premaster AI: mathematics for Machine Learning
IB4502 - Computer architecture
IB4602 - Ontology Engineering
IB4702 - AI and society
IB940U - Capita selecta
IB940V - Capita selecta
IB940W - Capita selecta
IB940X - Capita selecta
IB940Y - Capita selecta
IB940Z - Capita selecta
IB950A - Capita selecta

IB950B - Capita selecta
IB950C - Capita selecta
IB950D - Capita selecta
IB950E - Capita selecta
IB950F - Capita selecta
IB950G - Capita selecta
IB950H - Capita selecta
IB950I - Capita selecta
IB950J - Capita selecta
IB950K - Capita selecta
IB950L - Capita selecta
IB950M - Capita selecta
IB950N - Capita selecta
IB950O - Capita selecta
IB950P - Capita selecta
IB950Q - Capita selecta
IB950R - Capita selecta
IB950S - Capita selecta
IB950T - Capita selecta
IB950U - Capita selecta Bachelor's Computer Science
IB950V - Capita selecta Bachelor's Computer Science
IB950W - Capita selecta
IB950X - Capita selecta
IB950Y - Capita selecta
IB950Z - Capita selecta
IB9902 - Graduation preparation
IB9906 - Graduation project
IM0004 - Computer Science graduation assignment preparation
IM000S - CS Graduation Assignment Preparation
IM000T - CS Graduation Assignment Preparation
IM000V - Computer Science graduation assignment preparation
IM000W - Computer Science graduation assignment preparation
IM000X - Computer Science graduation assignment preparation
IM000Y - Computer Science graduation assignment preparation
IM000Z - Computer Science graduation assignment preparation
IM0203 - Software Architecture
IM0502 - Software Engineering graduation assignment preparation
IM050V - Software Engineering graduation assignment preparation
IM050W - Software Engineering graduation assignment preparation
IM050X - Software Engineering graduation assignment preparation
IM050Y - Software Engineering graduation assignment preparation
IM050Z - Software Engineering graduation assignment preparation
IM0603 - Design for change
IM0703 - Software quality management
IM0712 - Key Topics in Artificial Intelligence
IM0802 - Responsible Artificial Intelligence

IM0813 - Software Security
IM0823 - Software security
IM0902 - Bayesian Reasoning and Learning
IM0903 - System verification and testing
IM1002 - Machine Learning
IM1012 - Machine Learning
IM1102 - Deep Neural Engineering
IM1112 - Deep Neural Engineering
IM1202 - Model-Based Artificial Intelligence
IM1312 - Research Methods for Artificial Intelligence
IM1313 - Information Security Management
IM1412 - Generative Artificial Intelligence
IM1903 - Digital Governance
IM2203 - Software Engineering and AI
IM2303 - Security and AI
IM2403 - Software design and architecture
IM9502 - AI Graduation Assignment Preparation
IM9506 - Artificial Intelligence Graduation Assignment
IM9605 - CS Graduation assignment preparation
IM9606 - CS Graduation assignment preparation
IM9703 - SE Graduation assignment preparation
IM9906 - Software Engineering graduation assignment
IM990C - Computer Science graduation assignment
MAAI-2026-2027 - Master's programme in Artificial Intelligence
MACS-2026-2027 - Master's programme in Computer Science (MSc)
MAKSE - Master's programme Software Engineering, classroom-supervised
MASE-2026-2027 - Master's programme in Software Engineering (MSc)
MODL01 - Developing and delivering online and blended learning
MODL02 - Developing and delivering online and blended learning
OBAINF-2026-2027 - Open Bachelor's programme in Computer Science
SMAAI-2026-2027 - Bridging programme Artificial Intelligence
SMACS-2026-2027 - Bridging programme Computer Science
SMASE-2026-2027 - Bridging programme Software Engineering

BTI-2026-2027 - Bachelor's programme in Computer Science (BSc)

Dutch name: Bachelor Informatica (BSc) **Type:** Training **Language:** Dutch **ECTS:** 180 **Price:** -

How does the internet work? What exactly happens when you view a web page or send an email? How safe is it when you make a payment via a secure website? What is a database, how do you design such a database and how do you ask it questions? Why is logic so important for computer scientists? Why do computer scientists often think in terms of objects and classes? How can computers understand language and what role do formal descriptions play in this? How does a computer's operating system work? When does software have sufficient quality? These are questions that are addressed in computer science.

Set-up

The Bachelor's programme is general and broad in design. The first year (propaedeutic phase) is the introduction, and in the post-propaedeutic phase you deepen your knowledge and insight. In the courses Introduction to computer science and Introduction to information science, you get a broad overview of both fields, you get an idea of what you can expect from the programme and you become acquainted with studying at the OU. The core of the Bachelor's programme consists of coherent learning pathways in software technology, information systems and communication technology, with which you bring your subject knowledge to a good level. Computer scientists often work in an organisational and social context. You acquire the knowledge, understanding and ability to express this in daily work in several courses. You also take subjects in mathematics and theoretical computer science. You can also choose to broaden your studies, for example towards management sciences or psychology. This can be done via the so-called 'free elective space'. The Bachelor's programme has various integrating projects, including the Practicum design and implementation, which concludes the first year, and the Graduation project Bachelor's Computer Science, which concludes the entire Bachelor's programme. Each academic year, we organise study days for our computer science students. Would you like to know more about the content? Then request the study guide.

CPPAI - CPP Certified Software Architect

Dutch name: CPP Gecertificeerd Softwarearchitect **Type:** Standard product **Language:** Dutch **Price:** € 5,495.00

Artificial intelligence is a multidisciplinary field of psychology, cognitive science, statistics and above all computer science. In this programme, you become acquainted with the most important techniques underlying the success of AI applications from a computer science perspective. You learn about the most important AI algorithms and how they work (!), and also learn how they can be applied to solve practical problems. The study material is largely in English, but the programme is in Dutch.

CPPITSE - IT security engineer

Type: Standard product **Language:** Dutch **Price:** € 6,495.00

Why are there so many IT security problems these days? What are the underlying causes? How can software security be improved? In this academic-level programme, you learn to take security into account in all phases of a software development process. Explicit attention is also paid to the security of web applications. The programme has a practical approach and also offers sound theoretical depth.

In the IT security engineer programme you gain a broad overview of IT security and learn in particular to take security into account in all phases of a software development process. The emphasis is on the technical aspects of security. Attention is also paid to the broader framework of economic, organisational, ethical and privacy aspects.

The IT security engineer programme is a thorough academic-level programme that teaches IT professionals to understand security problems and to propose solutions. The programme first gives a broad overview of IT security, including network security and the security of web applications. In addition, we devote considerable attention to software security, applying security-by-design in all phases of the software development process. The motto is therefore: it is better to prevent security problems than to fix them! The programme offers sound theoretical depth at academic level, combined with practical sessions and attention to current security issues. Acquiring knowledge and skills go hand in hand. Sufficient time is built in to process and master the complex material.

FOICDI-2026-2027 - Focus programme Cybersecurity and Digital Innovation

Dutch name: Focusprogramma Cybersecurity en Digitale Innovatie **Type:** Training **Language:** Dutch
ECTS: 15 **Price:** -

This programme consists of the following courses:

1. Information security: a socio-technical approach

In this course, you learn that information security is not only a task of the IT department, but an integral part of business operations. You look at security from a technological, organisational, legal and social point of view.

- Focus: Organisational management of information security, risk analysis and the importance of the human factor.
- Result: You can independently, systematically and in a structured way develop control activities aimed at securing information.

2. Ethics in digital innovation

Not everything that is technically possible is also desirable. In this course, you reflect critically on the impact of digital technology on society.

- Focus: Values such as privacy, autonomy and justice. You learn to apply ethical perspectives to concrete digital innovations.
- Result: You can assess the impact of digital applications on stakeholders and proactively propose solutions to ethical challenges.

3. Security and IT

Here you delve deeper into the technology. You learn the principles of cryptography, authentication protocols, access control and the security of operating systems, software applications and computer networks.

- Focus: Why are IT systems vulnerable to cyberattacks and which measures exist to prevent and detect cyberattacks.
- Result: You understand the technical foundations of IT security and can apply this knowledge to make systems more robust.

No IT background? Start with the basics

If you do not yet have sufficient technical prior knowledge in the field of computer science, do not worry. Before the focus programme, you can optionally take the course Introduction to computer science. In this course, you become acquainted in an accessible way with the basics of programming, relational databases, operating systems, computer networks, cryptography and IT security.

IB0102 - Introduction to Computer Science

Dutch name: Inleiding informatica **Type:** Standard product **Language:** Dutch **ECTS:** 5 **Price:** € 410.00

The course discusses, in an introductory way, a number of diverse subjects from the field of computer science. Both classic subjects, such as relational databases, logic, binary arithmetic and operating systems, and modern subjects, such as object-oriented analysis, system design and programming in Python. Of course, the internet is discussed extensively, with special attention to its security. The subjects covered are presented in some depth. In this way, you 'taste' something of the field in a realistic way. You learn to make a simple database design, to encrypt a text and how mail programs send emails. You learn to read and write simple programs in Python, a popular language that is suitable for beginners but is also used for complex, large-scale projects. Because of the breadth of subjects, the course gives a good picture of the field of computer science and also of the Bachelor's programme in Computer Science. This course is therefore suitable both for those who want to know more about the field and for those who are considering taking the Bachelor's programme in Computer Science.

Learning objectives

After studying this course, you have elementary knowledge of a large number of sub-fields of computer science, including:

- programming: reading and writing simple programs;
- designing relational databases;
- (object-oriented) design and modelling of information systems;
- set theory and number theory;
- formal methods of description, such as predicate logic;
- computer operating systems;
- computer networks, communication via those networks and the internet;
- security of computer networks;
- symmetric and asymmetric cryptography.

IB0203 - Premaster programming techniques 1: relational databases and object-oriented programming

Dutch name: Premaster programmeertechnieken 1: relationele databases en objectgeoriënteerd programmeren **Type:** Standard product **Language:** Dutch **ECTS:** 7.5 **Price:** € 594.00

The Pre-Master's Programming techniques 1 consists of parts of the regular courses Relational databases and Object-oriented programming. A selection of the subjects covered: structure and rules of relational databases, joins and subselects, creating, managing and changing (the content of) a database; syntax and semantics of (parts of) Java, object orientation, working with the API, Strings, ArrayLists and arrays.

IB0302 - Relational databases

Dutch name: Relationele databases **Type:** Standard product **Language:** Dutch **ECTS:** 5 **Price:** € 410.00

In the early years of automation, a database was nothing more than an automated card index. The card indexes in the form of files were automatically accessible but each stood on its own. In the 1960s, it became possible to combine these different files with each other. A computer in a library, for example, is able to show not only which books there are, but also whether they have been lent out and, if so, to whom, where that person lives and when the book has to be returned. But a database system does more. It checks the correctness and completeness of the data and protects the data against unauthorised use. Building and querying such databases is central to this course. The course is a good theoretical and practical introduction

to the field. The 'relational databases' that are so important in practice are dealt with extensively, as is the underlying relational theory. You practise with the relational data language SQL, covering both querying and building and managing this type of database. In addition, there is special attention to normalisation of data structures, mechanisms for rule enforcement (including via 'triggers and stored procedures'), transactions, concurrency, optimisation of queries and the way in which a relational system administers its own structural data as 'metadata' in a system catalogue. The course consists of four blocks. Block 1 deals with relational database systems, elements of the relational model (including nulls and three-valued logic) and normalisation theory. Normalisation is not treated here as an information analysis method or technique, but as a suitable means of bringing a few important concepts to the fore (such as redundancy, functional dependency and 'single point of definition'). NB It is not a modelling course; existing database models are always assumed. Blocks 2 and 3 deal with querying, changing, creating and managing relational databases. SQL3 and the relational database management system Firebird are used for this. As far as possible, the course is lifted above learning a language: not only is relational theory continually brought forward and deepened, but also aspects that are important in all architectures (in particular navigation, in close relation to the methodical construction of queries). Block 4 deals with transactions and concurrency, rule enforcement via triggers and stored procedures and the data dictionary. The course is intended for anyone who wants to become proficient in building and querying relational databases, both by laying a good theoretical foundation and by practising a great deal.

IB0313 - Premaster programming techniques 2: object-oriented analysis, design and programming

Dutch name: Premaster programmeertechnieken 2: objectgeoriënteerd analyseren, ontwerpen en programmeren **Type:** Standard product **Language:** Dutch **ECTS:** 7.5 **Price:** € 594.00

The Pre-Master's Programming techniques 2 consists of parts of the regular courses Object-oriented analysis and design and Advanced object-oriented programming. A selection of the subjects covered: use cases, domain models, system sequence diagrams, UML interaction and class diagrams, GRASP; specifications, inheritance and polymorphism, abstract classes and interfaces, testing and exceptions.

IB0402 - Logic, sets and relations

Dutch name: Logica, verzamelingen en relaties **Type:** Standard product **Language:** Dutch **ECTS:** 5 **Price:** € 410.00

In computer science, logic is omnipresent. It is used as building blocks in the architecture of the computer, which consists of billions of transistors. Programmers then use it to control the computer at a higher level with programming languages. And finally, a computer program often has to meet certain specifications, which may or may not be formulated logically. In addition, more and more tasks are being taken over by artificial intelligence (AI), and for this too we must be sure that decisions are made in the right way. The first block of this course is therefore devoted to logic. A key concept in it is 'logical consequence'. In this course, we give a mathematical way of checking whether a statement follows from the assumptions. In the course, you learn to translate statements into logical formulas, in order then to analyse and understand them. Besides logic, in the second block of this course we deal with basic concepts from discrete mathematics. This is about describing sets, relations and functions. Although you may be used to calculating in mathematics, mathematics here does not focus on calculating and outcomes. The focus in this course is on the 'mathematical language' for formulating problems exactly, so that relations and properties can be described briefly and precisely. The course is concluded with a block on induction and recursion. These are mathematical concepts that are used to work with infinite sets. The set of numbers is infinite, but so are many sets you encounter in computer

science. Think, for example, of all possible inputs of a program. Induction provides a systematic approach for dealing with such sets.

Learning objectives

After studying this course, you can:

- translate statements into propositional and/or predicate logic;
- check whether a logical formula is valid or not, using various methods such as truth tables, laws and mathematical proofs;
- explain common concepts from discrete mathematics, such as set, Boolean algebra, graph, relation, function, equivalences and orderings;
- reason about, and check properties of, mathematical structures, such as graphs and functions;
- be familiar with inductive definitions and be able to give (simple) inductive proofs.

IB0602 - Linear algebra and stochastics

Dutch name: Lineaire algebra en stochastiek **Type:** Standard product **Language:** Dutch **ECTS:** 5 **Price:** € 410.00

Part 1 of the course is about linear algebra. We look at how we can solve systems of linear equations. We also deal with the representation of systems in matrices, after which we examine how we can work with matrices and what they can be used for. We end this part with two applications of linear algebra. We look at linear optimisation problems, which occur in logistics, and solve them with the simplex method. We then deal with the basics of computer graphics using linear algebra. Part 2 of the course is about stochastics. It covers how we can organise and present collected data. We offer you a mathematical basis for probability models and introduce various probability distributions. You learn how to estimate and test parameters for the probability distributions. This part, too, is concluded with an application: queues. Many processes can be modelled with queues. Questions about how long a treatment takes, or how likely it is that a process is blocked, we answer with stochastics.

Learning objectives

After studying this course, you have:

1. theoretical knowledge and skills for solving mathematical problems on the following subjects:

- linear equations, solution methods and Gaussian elimination;
- vectors and matrices and the relationship with linear functions;
- linear optimisation, the simplex method;
- combinatorics and probability theory;
- stochastics (random variables, expectations and variance);
- probability functions and probability distributions;
- statistics (descriptive statistics, regression, hypothesis testing);
- queues (counter models and blocking functions).

2. practical skills for setting up and analysing simple mathematical models for computer science problems;

3. (basic) general mathematical skills, such as drawing up a mathematical proof or reducing a mathematical problem step by step.

IB0702 - Computer networks

Dutch name: Computernetwerken **Type:** Standard product **Language:** Dutch **ECTS:** 5 **Price:** € 410.00

The course begins with a general introduction to computer networks and the internet. The most important concepts addressed are latency, throughput, abstraction and encapsulation. Reducing latency and increasing throughput are often necessary, simply because we want to improve the performance of a computer network. In addition, encapsulation and abstraction are ways of taming the complexity of systems and problems. Encapsulation is a common way of hiding implementation details. In computer networks, encapsulation is used to hide implementation details step by step. To achieve this, the functionality within a computer network is divided into five layers:

- application layer;
- transport layer;
- network layer;
- link layer;
- physical layer.

We also call this layered model the Internet protocol stack. The physical layer mainly describes how bits can be sent in a physical/electrotechnical way via different media. This layer is not covered in this course. The other four layers are each studied separately. The application layer is the top layer in the protocol stack, and it is the part of the network that you deal with directly as a user. Everyday things such as requesting a web page and sending an email are done using a protocol at the application-layer level. We pay attention here to the different communication architectures, such as client-server, and to the analysis of throughput and latency of a network connection. Every message at the application layer is sent via the underlying transport layer. There are roughly two ways of doing this, depending on the requirements that the application layer imposes. A connection can be set up in advance at the transport layer, which ensures that all messages arrive guaranteed and in the right order; this is done by the TCP protocol. If it is not necessary for all messages to arrive (for example when streaming a video), UDP can be used. Both protocols are discussed extensively, and we address the advantages and disadvantages of these protocols. Now that we can send messages at a high level from one computer to another, we can ask ourselves how to ensure that a message from Heerlen manages to reach a computer in the US. That is taken care of by the network layer using IP addresses. The network layer has two responsibilities. At local level, a forwarding table is used to determine to which ‘neighbouring system’ a message must be forwarded. This happens mainly in routers: nodes that link different parts of the network to each other. At global level, it must be ensured that this forwarding table is filled in correctly, so that via a number of hops (steps along various routers) the message ultimately [...]

IB0802 - Formal languages and automata

Dutch name: Formele talen en automaten **Type:** Standard product **Language:** Dutch **ECTS:** 5 **Price:** € 410.00

Languages can be natural languages (Dutch, English), but equally programming languages such as the Java language. Anyone who wants to translate a sentence from, for example, Dutch into another language, such as English, needs thorough knowledge of both the structure (syntax) and the meaning (semantics) of the two languages. If a translation is to be carried out by a computer, the syntax and semantics must be laid down formally in some way. For natural languages, that is a very difficult matter. Formal languages, such as programming languages, have a much simpler structure, which can be checked automatically. That structure is laid down in a grammar or in an automaton. In this course, various kinds of formal languages with the associated grammars and automata are addressed. Attention is also paid to designing these grammars and automata yourself. The first block forms a general introduction to formal languages. In the second block, the properties of regular languages and finite automata are dealt with. Context-free languages and pushdown

automata are the subject of block three. In the last block, Turing machines, the Chomsky hierarchy and the concepts of decidability and complexity of problems are addressed.

IB0902 - Advanced object-oriented programming

Dutch name: Geavanceerd objectgeoriënteerd programmeren **Type:** Standard product **Language:** Dutch
ECTS: 5 **Price:** € 410.00

The course consists of four blocks. The largest part of the first block is devoted to the question of what a good program is and to the type system of Java. We describe the requirements that a good program must meet in the form of specifications, which form the basis for the implementation and for developing associated test cases. Inheritance is dealt with as part of the type system. Dynamic binding of methods is a key concept here. Abstract classes and interfaces are then addressed, concepts that play an important role in defining so-called design patterns: schematic solutions to common problems. The block ends with a brief treatment of Generics: the extension of Java that makes it possible to use type parameters. The second block is about kinds of errors that can occur and how they can be handled, for example by means of exceptions. Debugging and testing are briefly addressed, among other things. Block three offers an introduction to programming with threads, which makes concurrency possible. We address what concurrency exactly is and show how a concurrent program can be made using Java's Executor Framework. Concurrent programs are complex and can contain subtle errors. We therefore describe how you can analyse a simple concurrent program. The last block deals with the construction of user interfaces. Two learning units address working with Swing components; the event-handling mechanism is also covered. The last learning unit addresses how the cohesion and coupling of OO programs can be improved and, as an application of that, deals with the Observer pattern, a widely used and important design pattern that makes it possible to keep the domain layer independent of the user interface, and the Model-View-Controller pattern, a widely used architecture for separating responsibilities.

Learning objectives

After taking this course, you can:

- interpret a contract for a class and method;
- provide an implementation on the basis of a contract and draw up associated tests;
- understand and apply inheritance (including the use of abstract classes and interfaces), exception handling and threads;
- define simple generic classes;
- explain how programming to an interface can be used in object-oriented programs;
- program a simple concurrent program;
- program a graphical user interface yourself using the package javax.swing;
- assess a program for cohesion and coupling and improve that program;
- apply the MVC model and the Observer pattern.

IB1002 - Object-oriented analysis and design

Dutch name: Objectgeoriënteerd analyseren en ontwerpen **Type:** Standard product **Language:** Dutch
ECTS: 5 **Price:** € 410.00

Information systems meet the information needs of organisations, people or installations/systems. Examples are the mortgage system at a bank, an internet ordering system of a book and record shop, the autopilot in an aircraft, the journey planner of the Dutch Railways (NS) and a library's online catalogue. Developing information systems takes place in a dynamic playing field of methods that emerged in practice, wilful developers, academic research results, standardisation initiatives, commercial tools and a stubborn reality. Developing and building an information system calls for a well-considered, project-based approach, in which a

development team is confronted with evolving insights among various stakeholders such as the client, the client's customers and diverse users. The development process has a number of characteristic activities (drawing up requirements, analysis, design, construction, implementation). The course pays attention to the entire development process, from requirements to implementation, but the emphasis is on the analysis and design of the information system. Other aspects (implementation, testing, project management) are dealt with in less depth. Subjects addressed include:

- formulating requirements and, resulting from that, the specification of the information system;
- iterative system development as it takes place in the context of, for example, the Unified Process method;
- the Unified Modeling Language (UML) as a language in which to represent design and analysis models;
- drawing up a domain model;
- learning to design the message traffic between objects (in the form of interaction diagrams) on the basis of a number of guidelines, the so-called GRASP guidelines;
- drawing up a design class diagram on the basis of the interaction diagrams.

The course is intended for anyone who wants to become proficient in designing object-oriented information systems. The course is not a UML course, but a course in which you learn to design using a number of UML diagrams.

Learning objectives

After taking this course, you can:

- explain how the Unified Process proceeds, which artefacts can be produced in this process and how iterations are planned in this process;
- for a small information system, on the basis of a case description and unstructured material from specifications, model the desired behaviour of that system using use case diagrams, use cases, system sequence diagrams, activity diagrams, state diagrams and a domain model;
- indicate what the characteristics of a good object-oriented design are and what role patterns play in it;
- for a given use case from a limited information system, draw up an object-oriented design in the form of interaction diagrams and a design class [...]

IB1102 - Object-oriented programming

Dutch name: Objectgeoriënteerd programmeren **Type:** Standard product **Language:** Dutch **ECTS:** 5
Price: € 410.00

Block 1 is an introduction of four learning units, which alternately emphasise the Java language and the Eclipse development environment. In this block, you become acquainted with object orientation by using existing classes. First, you develop a very simple Java program and run it with minimal tools, so still without a development environment. You then learn how to write and process programs within the Eclipse development environment. Next, you learn to write programs that use simple forms of selection and repetition. Finally, you learn how to use Eclipse to give the programs a graphical user interface. In block 2, you learn everything you need to know for this course about object orientation. This begins with an introduction at a conceptual level, so (largely) independent of the Java language. In this learning unit, the concepts 'class' and 'object', among others, are explored in depth and the relationships between classes are clarified. The concept of 'inheritance', which is important in OO, is also addressed here for the first time. You then learn to define classes yourself. You also learn to work with the extensive Java library and its documentation on the internet. The last learning unit of the block is about the way objects are represented in memory and the consequences of this. Block 3 deals with the implementation of methods (these are in a certain sense the most important program units, because this is where the programmer codes the desired behaviour of the objects). Because of the cyclical approach followed, we had already anticipated this in block 1. In this block, all

elementary statements are treated in detail: primitive types and expressions, selection statements, repetition statements and working with Strings and lists (arrays and ArrayLists). The block also pays attention to documenting and testing (JUnit).

Learning objectives

After taking this course, you can:

- independently design, from a given specification, a simple object-oriented program consisting of a limited number of classes, in which each class has a clear, limited responsibility;
- use the syntax and semantics of (parts of) the Java programming language to implement the design in a working program;
- provide a program with a simple graphical user interface;
- use memory models to explain how a program works;
- design and implement simple algorithms;
- write clear and understandable Java program code;
- document classes with Javadoc;
- write test programs to test the correct working of a class.

IB1202 - Practicum design and implementation

Dutch name: Practicum ontwerpen en implementeren **Type:** Standard product **Language:** Dutch **ECTS:** 5 **Price:** € 410.00

The practicum consists of a case description, followed by five tasks. The case is about developing a management information system for a nationally operating organisation, Prik2Go, where people can go for vaccinations. With the help of the management information system, it must be possible to investigate whether branches should be closed or opened. Task 1 consists of a number of questions in which you are asked to reflect on the relevant material from the prior-knowledge courses Object-oriented programming, Object-oriented analysis and design and Advanced object-oriented programming. This task is intended to refresh knowledge from these first-year courses and to prepare you for subsequent tasks. In task 2, you use a use case to design part of the management information system. Specific functionalities of the system are analysed and modelled according to the Rational Unified Process (RUP). In task 3, you analyse the existing data using SQL. This is an application of the material from the subject Relational databases. You specify (complex) queries yourself and implement and test them. In task 4, you implement in Java the part of the system that you designed in task 2. In task 5, you work on a second use case. As in task 2, you again make an analysis and a design and then implement the use case in Java.

Learning objectives

The aim of this course is that you apply knowledge, insights and skills that you acquired in various first-year courses in an integrated way to a case. After studying this course, you can:

- draw up complex queries to query a database;
- demonstrate that you have the following skills: analysing, modelling, designing, programming and evaluating a small object-oriented information system with an underlying database according to a (R)UP approach, from drawing up the functional specifications through to delivery, including testing the software;
- complete tasks within a set time.

In addition to integrating knowledge from the courses in which analysing, designing and programming are central, in this practicum you pay attention to the quality of software. We ask you to provide your code with documentation, to ensure readable code (for example by using meaningful names, writing short methods and

including helper methods), and to test, both at the level of classes by means of JUnit tests and at the level of a whole application by means of integration tests.

IB1302 - Operating systems

Dutch name: Besturingssystemen **Type:** Standard product **Language:** Dutch **ECTS:** 5 **Price:** € 410.00

Every modern computer is controlled by an operating system. Well-known operating systems are Windows, macOS and Linux, but operating systems such as Android and iOS also run on phones. The operating system comprises the software that provides the interface to users and that regulates access to the available hardware for programs. In the course, the emphasis is on managing processes, memory, files and peripherals. After a general introduction, the concepts ‘process’ and ‘thread’ are dealt with. These are the basic units for running software simultaneously. Every program that is run consists of at least one process, and within a process several threads are often used (example: with video, sound and image are often decoded simultaneously in separate threads). Concepts such as ‘parallelism’ and ‘concurrency’ play a major role here. Using these building blocks, we look at all kinds of possible problems when running software at the same time and when software cooperates, such as deadlock, livelock and starvation. Another aspect that the operating system manages is memory. As a rule, the physically present memory (addressed by physical memory addresses) is too small to accommodate the operating system itself, the other system software and all user applications. That is why most systems make use of virtual memory (addressed by logical memory addresses), which can be many times larger than the physical memory and which makes use of secondary storage (usually a hard disk or flash memory). Parts of programs can be placed in physical memory and parts in secondary storage. The hardware support needed to make the modern memory system function correctly and quickly is also covered. Programs and data are stored as files on the hard disk. A mechanism is also needed for this to create, edit and delete files. This is the file management system. The file management system also takes care of the logical file storage structure that uses folders, and of its translation to the physical storage on secondary storage. Finally, the input-output system is dealt with. Without such a system, the computer cannot communicate with the outside world. Peripherals such as the monitor, keyboard, mouse and printers must be able to be connected. However, internal components of the computer also use the input-output system. Think of disks such as the graphics controller. Here too, knowledge of the underlying processor architecture is essential. To apply the knowledge of operating systems in practice as well, a practicum is part of the course Operating systems. In this practicum, we use a mini operating system for Raspberry [...]

IB1402 - Communication skills

Dutch name: Communicatievaardigheden **Type:** Standard product **Language:** Dutch **ECTS:** 5 **Price:** € 410.00

Computer scientists and information scientists need good communication skills to carry out their work successfully. Career opportunities are determined not only by subject knowledge of a field; other skills are certainly just as important. In this course, conversation, writing and presentation skills are addressed. The course consists of two parts. In the part ‘Conversation skills’, the theory of oral communication is dealt with first. Then, during two meetings, this theory is applied in a number of practice conversations between students. You learn to observe and assess your own conversations and to formulate points for improvement in order to develop your conversation skills further, both during and after the course. Both meetings are concluded with a practice conversation about which you write a reflection report.

Learning objectives

After completing the part Conversation skills, you:

- have knowledge of the theory of oral communication;

- can apply this theory in business conversations;
- can observe business conversations of others, give feedback on them and formulate advice for improvement;
- can reflect on your own contribution to business conversations and formulate advice for yourself to improve it.

In the second part, 'Writing and Presenting', you write an article on a subject of your own choice. A methodical approach to the writing assignment is paramount. There is also attention both to communication between IT specialists and to communication between IT specialists and non-specialists, and what these differences mean for the skills required. Halfway through the writing process, you give a presentation on the chosen subject.

Learning objectives

After completing the part Writing and Presenting, you:

- have knowledge of the writing process and how to approach this process methodically;
- have knowledge of the requirements for business texts;
- can make an analysis of a writing situation;
- can write a well-structured text that fits the target group.

In addition, you:

- have knowledge of the requirements for a good presentation;
- can make a well-structured and understandable presentation;
- can give a presentation for a small group of fellow students.

IB1502 - Data structures and algorithms

Dutch name: Datastructuren en algoritmen **Type:** Standard product **Language:** Dutch **ECTS:** 5 **Price:** € 410.00

To be able to program well, a number of skills are important. First of all, you must be familiar with the language constructs of the programming language in which you program, and you must be able to structure your program in a logical and clear way into sub-components. These skills receive ample attention in the courses Object-oriented programming and Advanced object-oriented programming. In those courses, however, relatively little attention is paid to algorithms and the data structures on which these algorithms work. Think of methods and techniques for storing data, looking it up quickly, sorting it, or finding an optimal route in a network in an efficient way. This course gives an overview of the data structures with which data can be recorded and of the associated algorithms. After three introductory chapters on design principles and analysis techniques, the basic data structures (and the algorithms operating on them) from computer science are dealt with in nine chapters: stacks, queues, sequences, (search) trees, priority queues, sets and (weighted) graphs. Attention is also paid to recursion. The algorithms are in many cases presented in language-independent pseudocode; in addition, (part of) the implementation in Java is regularly shown. Besides the treatment of the algorithms themselves, considerable attention is also paid to the analysis of algorithms, in particular to be able to predict how much time an algorithm takes for a problem of a particular size. The same kind of analysis is used for determining the space requirements of the data structures on which the algorithms work. A somewhat mathematical approach is necessary for these analyses. At the end of the course, you will have gained an overview of, and insight into, the standard data structures and algorithms.

IB1602 - Functional programming

Dutch name: Functioneel programmeren **Type:** Standard product **Language:** Dutch **ECTS:** 5 **Price:** € 410.00

The course uses the second edition of the English-language textbook Programming in Haskell by Graham Hutton. The programming language is introduced step by step and the most important concepts of the language are dwelt on. Topics include recursive functions (a function that calls itself), higher-order functions (a function that has another function as a parameter) and functions in which pattern matching is used to distinguish cases. The principle of currying and the partial application of functions with several parameters are dealt with. Various functions on lists are also covered and list comprehensions are introduced; this is a special language construct for writing down calculations with lists in a short and intuitive way. Haskell has a powerful type system that statically checks the types of a program without running this program. With this type system, many kinds of errors in a program can be detected early. All types of a program can be inferred automatically, which means that the number of mandatory type annotations for a programmer is minimal. Even so, it is good practice to write down function types explicitly. Parametric polymorphism (comparable to Java Generics) and overloading of functions via type classes are addressed. You also learn to define new types yourself via datatype declarations and type synonyms. In the later learning units, some more advanced subjects are covered. For example, the way in which input-output can be realised in a language in which side effects, such as writing a file or reading a keystroke, take place in a special way is examined. In Haskell, this is done with the mathematical concept of the monad. It is also studied how expressions are only evaluated at the last moment (lazy evaluation) and why this makes it possible to work with infinite data structures. Finally, the principle of induction is explained, for reasoning about a program and for drawing up simple proofs. In addition to the textbook, there is an online learning environment in which the most important subjects are summarised once more, difficult subjects are explained further and in which there is a wealth of assignments to practise the material yourself. The course has a special requirement consisting of two practicum assignments. Both assignments must be completed with a pass. These assignments make it possible to start work yourself on the subjects studied. The course is concluded with a written open-book exam on the whole material of the course.

IB1812 - Security and IT

Dutch name: Security en IT **Type:** Standard product **Language:** Dutch **ECTS:** 5 **Price:** € 396.00

The course starts with a general introduction in which concepts and principles in the field of IT security are explained. The course contains an introduction to cryptography, in which the basic principles of symmetric and asymmetric cryptography and applications are addressed. Ample attention is paid to authentication, with which users can prove their identity, as well as authentication protocols for exchanging a cryptographic key. Ample attention is also paid to the security of operating systems and software applications, and various forms of malicious software are explained. Finally, the course addresses the security of the internet. The (in)security of internet protocols and various security measures are discussed. The course contains various practicals. For example, you can simulate the workings of cryptographic algorithms. In a virtual laboratory, you can carry out attacks yourself as a hacker and take security measures.

IB2012 - Academic writing skills

Dutch name: Wetenschappelijke schrijfvaardigheden **Type:** Standard product **Language:** Dutch **ECTS:** 5 **Price:** € 410.00

Writing articles is an important skill. A well-written and clearly structured article can be a pleasure to read. To write such an article, you have to take the reader by the hand and convince them of the relevance, correctness and adequacy of the ideas. In this course, you learn to write in the academic writing style. First

you choose a subject from four proposed subjects. Then you carry out a literature scan on your chosen subject and subsequently write an article on it. As a last step, you act as editor of your own article.

IB2302 - Distributed algorithms

Dutch name: Gedistribueerde algoritmen **Type:** Standard product **Language:** Dutch **ECTS:** 5 **Price:** € 410.00

A distributed system is a software system that consists of several components that run on different machines and exchange messages with each other via a network. Distributed systems are indispensable in modern society: web applications, cloud technology and blockchains are just a few examples of distributed systems that both IT professionals and 'ordinary citizens' are confronted with daily. The demand for distributed systems will moreover only increase in the future, fuelled by new developments and innovations such as the internet of things and smart cities. Distributed systems have a number of characteristics that make their development extremely challenging. An important source of complications is the fact that message traffic between the components of a distributed system is not instantaneous but asynchronous: there is always some time between sending and receiving a message, and the sender moreover does not know when the receiver has actually processed the message. To make the situation even more complex, it is also possible for messages to be received in a different order from that in which they were sent, even if there is only one sender. The art is, despite these uncertainties, to nevertheless ensure that all components exchange information with each other correctly and jointly realise the desired functionality. In this subject, we study the fundamentals of distributed systems from an algorithmic perspective: we become acquainted with a number of distributed algorithms, whose execution is spread over the components of a distributed system, and in which communication between the components plays an essential role. An example of a fundamental problem is the following. Suppose that one of the components of a distributed system wants to make a copy of the global state of the entire system (namely, the values of all variables in all components). The problem may seem simple: the component simply sends a message to all other components asking them for a copy of their local state, after which those other components make such a copy and send it back. The problem is the timing: the time between sending and receiving the copy requests is different for each of the components, so the copies are not all made at the same moment. The consequence is that the collection of local states that is ultimately delivered has not necessarily been a global state of the entire system, but a fusion of separate parts of a series of global states. Other fundamental problems that we will look at are: detecting a deadlock (all components are waiting for each other and consequently none of them makes progress); detecting termination (all components have finished their work); the choosing [...]

IB2602 - AppLab

Type: Standard product **Language:** Dutch **ECTS:** 5 **Price:** € 410.00

For the app, you use modern techniques, as customary at the time (currently including: reactive, real-time distributed database, view/data binding and web services). You work in groups on the app, using a modern source code system (GitLab). In AppLab, you also make active use of Continuous Integration (CI) and Continuous Delivery (CD), so that the code produced can be tested and distributed automatically. We use two ways of working from the start-up world: 'release early, release often' (to get feedback quickly and adjust your plans) and 'fail early, fail fast' (realise the riskiest part first so that you know whether it makes sense to continue). You receive active feedback from the teachers on the code and way of working through the peer review functionalities of the source code system used. You start with an already set-up source code system environment with a given basic app with the techniques mentioned. In this way, you can get started quickly and concentrate on realising your own app. In AppLab, you learn in practice to make a modern app in a time-efficient way.

Learning objectives

After studying this course, you can:

- work in a team with the support of a modern source code management system;
- efficiently evaluate and use new techniques, frameworks and libraries;
- develop 'modern' apps from front to back and from start to finish;
- achieve and safeguard code quality in a practical setting.

IB2702 - Concepts of programming languages

Dutch name: Concepten van programmeertalen **Type:** Standard product **Language:** Dutch **ECTS:** 5 **Price:** € 410.00

Programming can be done in many different ways, in different styles. Each programming style has its own specific programming languages. For example, we distinguish imperative, object-oriented, parallel, functional and logic languages. Each language in turn has its own advantages and disadvantages in diverse respects such as expressive power, availability of implementations, reliability, efficiency and theoretical foundation. The object-oriented programming style has been covered in the courses Object-oriented programming and Advanced object-oriented programming. The functional language Haskell is covered in the course Functional programming. You may already have become acquainted with the logic programming style. The parallel programming style is discussed extensively as part of the course Concepts of programming languages. Aspect-oriented programming is also addressed. The main aim of this course is to provide insight into the differences between the various languages. An important aspect of this course is that the emphasis is much more on programming language concepts than on programming itself. Examples of language concepts are the data types in a language, the kinds of variables and statements, and the different possibilities of abstraction. In modern programming languages, several programming styles are offered side by side. A well-known example is the object-oriented language Scala, in which functional programming is also possible. In the course, we look at what this can mean for the programmer. This is addressed in the course in the Scala case study.

Learning objectives

After studying this course, you can:

- name and explain the basic concepts of programming languages, such as values, types, expressions, variables, statements, bindings and abstraction mechanisms;
- indicate to what extent a concrete programming language meets the four quality criteria for programming languages formulated in this course: the type completeness principle, the qualification principle, the abstraction principle and the correspondence principle;
- name and explain the encapsulation techniques, type systems and ways of interrupting program processing;
- name and explain the concepts of parallel programming and the problems that can occur with it;
- describe in your own words which language concepts are characteristic of imperative, object-oriented, functional, parallel, logic and scripting programming languages respectively.

IB2812 - Artificial intelligence

Dutch name: Kunstmatige intelligentie **Type:** Standard product **Language:** Dutch **ECTS:** 5 **Price:** € 410.00

Artificial intelligence is concerned with the underlying principles of intelligent behaviour in natural or artificial systems. This course treats artificial intelligence from the perspective of an 'intelligent computational agent'. Who or what exactly that agent is, a computer, a thermostat or something entirely different, remains open,

but the agent acts in and responds to an environment in an intelligent way. You will become acquainted with a number of the most important principles and techniques for developing computational agents, for example to solve search problems and planning problems, or to make agents reason about available knowledge. You also learn how an agent can learn from data by using modern machine learning algorithms.

Learning objectives

You become acquainted with a number of methods and techniques from artificial intelligence in the fields of searching, reasoning, learning and planning. After studying this course, you can:

- describe the structure of an agent system, and reason about the consequences of choices in a simple agent system;
- describe various search techniques (A*, iterative deepening, branch and bound) and apply them to solve a concrete search problem;
- formalise problems by means of features and constraints and use techniques to solve these formalised problems in a computationally efficient way;
- specify a Bayesian network by hand and calculate probabilities from it;
- give a description of a number of (supervised) learning techniques and the differences between them, and also apply them to a small dataset;
- apply neural networks, including working through the forward and backward procedure by hand for different activation functions;
- represent a planning problem in formal logic and solve it by means of planning algorithms.

IB2912 - Logic and computer science

Dutch name: Logica en informatica **Type:** Standard product **Language:** Dutch **ECTS:** 5 **Price:** € 410.00

Traditionally, logic is the study of correct reasoning. Even in classical antiquity, people were concerned with analysing the structure of reasoning. Even now, logic can be an important tool when we want to examine the correctness of reasoning. Logic is therefore applied in numerous sciences such as linguistics, philosophy and mathematics. An important recent field of application is computer science. The course consists of two blocks. In the first block, a formal language, propositional logic, is introduced with which the structure of reasoning can be made visible. For example, ‘where there is smoke, there is fire’ can be translated into s->f. A formal language only becomes meaningful once the meaning (semantics) of this language has been laid down. Propositional logic uses truth tables for this. The semantics enables us to define when a formula is a valid consequence of another formula. With the help of semantic tableaux, the validity of an inference can be checked in a fairly efficient way. In addition to this semantic approach, in this block you also learn to make derivations using an axiomatic system. To analyse statements in which objects, properties of objects and variables play a role, a richer language is needed: that of predicate logic. This language is central in the second block of the course. For predicate logic too, both the semantics and an axiomatic proof system are explained. The course deals with two applications of predicate logic within computer science. The first application is aimed at imperative languages. You learn how ‘Hoare calculus’ can be used to prove the correctness of programs. In the second topic, logic programming, you become acquainted with general resolution and theorem proving by means of skolemisation.

Learning objectives

After studying this course, you can:

- use semantic tableaux to check validity and consistency, and to examine whether a formula is a tautology;
- give axiomatic proofs in both propositional logic and predicate logic;
- provide proofs about logic using formula induction;

- give resolution proofs, and, using rewriting, Skolem forms and unification, also apply general resolution;
- for a STIP program, determine a precondition and a postcondition;
- prove the correctness of simple programs using Hoare calculus;
- interpret predicate logic formulas on models, and find models and counter-models for predicate logic formulas.

After studying the course, you have insight into the possibilities and limitations of the logical language and insight into the distinction between syntax and semantics.

IB3112 - Software engineering

Type: Standard product **Language:** Dutch **ECTS:** 5 **Price:** € 410.00

Software is all around us, from the smallest devices to worldwide systems. Software engineering is the field that deals with all aspects of producing high-quality software. Software that is realised by several people and has to function efficiently. Such software is often used for a longer time by several people, so that it must be adaptable and maintainable and be able to remain in operation sustainably at reasonable cost. In this subject, you study both the theory and the practice of software engineering. In the first part of the course, you read an interesting book about modern ways of designing and developing software. On that basis, you go deeper into important subjects by means of two assignments. In the second part, you work together with two fellow students on a project, in which all kinds of aspects of software engineering can be learned in practice. Thanks to the challenges that can be tackled at different levels, students can make their contribution in different ways, from project management to technical execution. In the project, you and your group develop a prototype further towards a software product that can be useful for a particular target group. The subject concludes with writing a professional reflection report on this project.

Learning objectives

After studying this course, you can:

- (software process, knowledge) characterise different software processes;
- (context of software, interpret) distinguish software products and software projects in relation to the goals of the customers and the developers;
- (agile, analyse) make a well-founded choice of agile approaches (including Scrum, Extreme programming) for all aspects of the software process;
- (new technical aspects, evaluate) learn and evaluate a new ICT technology (including software architecture, cloud-based system);
- (modern software development, carry out) make a contribution to an existing project using modern aspects and tools (including a potentially new programming language, Git, automated unit testing);
- (reliability, apply) apply software development techniques that increase the reliability (in terms of security, privacy, testing and avoiding errors, etc.) of the software;
- (collaboration, apply) collaborate in a small group on a software project using the Scrum methodology;
- (academic writing, apply) reflect on your own work and write down that reflection clearly.

IB3202 - Software testing

Dutch name: Software Testen **Type:** Standard product **Language:** Dutch **ECTS:** 5 **Price:** € 410.00

We are surrounded by software. From pacemaker and refrigerator to toothbrush, it is in everything. Our lives are increasingly determined and influenced by it. Guaranteeing the quality of software is therefore vital. Unfortunately, properly working software is still not a given. In fact, if we continue like this, we might come close to a software quality crisis. A crisis in which the lack of software quality costs more than the software

itself. To prevent that, we must create a culture of software quality thinking in which testing is central. This course gives an overview of the field of testing. We outline differences and similarities between exploratory testing and structured testing. We deal with all kinds of test models, from equivalent input partitions to much more formal state models. In this course, we do not distinguish between black-box and white-box techniques. The information we use to construct the models can be either black-box or white-box; the technique does not change because of that. Together with the coverage criteria for each model, we can develop test suites. We deal with combinatorial testing as an overarching technique for making combinations. We also look at mutation testing to determine the quality of test suites. Finally, we address managing an entire test process, what different components there are and how you can improve those processes through existing improvement models.

Learning objectives

After studying this course, you can:

- give a general overview of the field of software testing;
- understand the complexity, the challenges and the purpose of software testing;
- name, understand and draw up a number of models for software testing;
- for these models, name, understand and apply a number of techniques for designing test cases;
- for these techniques, name, understand and use a number of suitable coverage criteria;
- assess the quality of a test set;
- have a software quality mindset.

IB3502 - Premaster AI: programming with Python

Dutch name: Premaster AI: programmeren met Python **Type:** Standard product **Language:** Dutch **ECTS:** 5 **Price:** € 410.00

This course takes you at a fast pace through what the Python programming language has to offer. Python has handy data types such as lists, tuples, sets and dictionaries, and is a very suitable language for investigating large data files. Python is not a statically typed language. There is no compiler that checks a program for errors related to types. It is therefore extra important that, as a programmer, you avoid errors yourself. We therefore pay a lot of attention to ‘neat’ programming, to thinking before you start writing code, to documentation and to testing code. You begin with code of one or two lines that you can test using an interactive interpreter, from which you gradually learn to write larger programs using functions and classes. You also learn to organise your code in modules. We pay attention to how you organise your code so that it remains easily extensible. Python is a handy language for collecting data (for example via the internet), processing, investigating and organising that data, and storing it in databases. In this course, you learn how to do that, using tools such as Jupyter Notebook and pandas.

Learning objectives

After studying this course, you can:

- independently write a program from a given specification, using the data structures that Python offers, functions, classes and modules;
- provide a function, class or module that you have written yourself with documentation;
- test functions and programs and draw up test cases for them yourself;
- explain the difference between primitive types and non-primitive types;
- use the data structures that Python offers and explain the differences between them;
- write clear and understandable Python program code;
- collect data in various ways using Python;
- analyse data using Python and visualise its content for exploration.

IB3702 - Premaster AI: mathematics for Machine Learning

Dutch name: Premaster AI: wiskunde voor Machine Learning **Type:** Standard product **Language:** Dutch
ECTS: 5 **Price:** € 410.00

This course gives a deeper insight into linear algebra and an introduction to continuous mathematics. The first part addresses linear transformations, which are widely applied in machine learning and AI. Attention is also paid to the theoretical aspects of linear algebra, including vector space, eigenvalue and matrix decomposition. The second part gives an introduction to calculus (limits, differential and integral calculus). Calculus is applied everywhere in the physical world, but also often in AI, for example when expectations have to be expressed or an optimal solution has to be found. This subject uses the textbook 'No Bullshit Guide to Linear Algebra', an OU workbook and digital learning material. You also use the computer to carry out calculations, as happens in practice too. The aim of this subject is that after this course you can use the necessary mathematics well and with confidence in the Master's programme in AI.

Learning objectives

After studying the course, you have:

1. theoretical knowledge and skills for solving mathematical problems on the following subjects:

- vector space;
- norms, distances and orthogonality;
- eigenvalues and eigenvectors;
- linear transformations;
- sequences and series;
- limits and differentiation;
- function analysis;
- integral calculus.

2. practical skills for setting up and analysing simple mathematical models for AI problems.

IB4502 - Computer architecture

Dutch name: Computerarchitectuur **Type:** Standard product **Language:** Dutch **ECTS:** 5 **Price:** € 410.00

Most people have an intuitive understanding of what a computer can do, and how to deal with it. But how does a computer actually calculate, and how can we ensure that those calculations are carried out faster? In this subject, we look under the hood of a modern computer, and learn how programs are executed. We address the various components of a computer (processor, memory, input/output), and the role they play in executing a program. In addition, we pay attention to various methods that have been applied in recent decades to make computers faster, such as pipelining, branch prediction and caching. We discuss the underlying principles of computer hardware design, and the relevant calculation rules for analysing their performance. By understanding these subjects, after the course you will be able to reason about programs that make optimal use of these techniques and are therefore faster.

Learning objectives

After taking this course, you:

- can carry out simple calculations with binary numbers;
- understand how data and instructions are stored in a modern computer;
- can translate simple programs in a higher-level programming language into assembly code;
- can read and manipulate simple programs in assembly code;
- understand how assembly code is converted into machine code;
- can design simple digital circuits;

- understand how digital circuits execute machine code;
- can reason about the performance of a program;
- know which techniques hardware components use to improve performance;
- understand how program code can make optimal use of hardware components;
- understand the reasons for switching from sequential to parallel programs.

IB4602 - Ontology Engineering

Type: Standard product **Language:** Dutch **ECTS:** 5 **Price:** € 410.00

Ontologies were originally developed within artificial intelligence as a way of recording knowledge about the world in a structured way. Nowadays there are many applications outside artificial intelligence. For example, ontologies offer an elegant method for developing information systems: unlike relational data models, ontologies can be used to lay down the structure of data in a platform-independent way and to derive new knowledge from data. In addition, ontologies offer, among other things, a solution for achieving interoperability between systems. The course Ontology Engineering is about representing, designing and implementing ontologies to record information and knowledge. You become acquainted with description logic, a formal logic used for formulating and reasoning about ontologies. You also become acquainted with OWL, a language based on description logic that was developed for the web. In addition to these theoretical subjects, the practical side of ontology engineering is also covered extensively. For example, you learn methodologies for drawing up an ontology in a structured way, and you work with software tools to draw up and process ontologies.

IB4702 - AI and society

Dutch name: AI en maatschappij **Type:** Standard product **Language:** Dutch **ECTS:** 5 **Price:** € 410.00

AI is playing an ever greater role in companies and society, and the specific technologies follow each other quickly. Companies therefore need people who have a good overview of the field and can assess which techniques are relevant for their company. In the course AI and society, you study the workings and possibilities of modern AI techniques and the potential social problems that these techniques may or may not cause. Not only the possibilities, but also the limitations of AI are highlighted in this course. There is also a lot of talk in the media about the dangers of AI, but not all of them are equally well founded (for example, superintelligence does not exist at all yet). In this course, you gain a better understanding of AI and its actual dangers. The course is suitable for participants with both a technical and a non-technical background. In the first half of the course, you learn about various AI techniques; the second half focuses on the responsible use of AI.

Learning objectives

After completing this course, you can:

- name the most important techniques underlying AI systems;
- reason about the social consequences and acceptance of AI;
- independently make proposals on which AI techniques can best be used to solve specific problems;
- assess whether a business problem can be solved by a particular AI technique;
- reflect on the social and ethical consequences of applying AI;
- critically assess media coverage of AI;
- exchange opinions with colleagues and fellow professionals and give presentations about AI.

IB940U - Capita selecta

Type: Standard product **Language:** Dutch **ECTS:** 2.7 **Price:** € 218.74

You can register for this course if you have completed all other components of the Bachelor's programme in Computer Science or the Bachelor's programme in Information Science, or are enrolled for them, and you want to end up with exactly 180 credits for the entire programme. An individual assignment will be formulated for you, taking into account the number of credits of this course. Typically, the assignment is closely linked to your graduation assignment. This avoids a familiarisation period and offers the opportunity for further in-depth work on your graduation project.

IB940V - Capita selecta

Type: Standard product **Language:** Dutch **ECTS:** 2.1 **Price:** € 177.52

You can register for this course if you have completed all other components of the Bachelor's programme in Computer Science or the Bachelor's programme in Information Science, or are enrolled for them, and you want to end up with exactly 180 credits for the entire programme. An individual assignment will be formulated for you, taking into account the number of credits of this course. Typically, the assignment is closely linked to your graduation assignment. This avoids a familiarisation period and offers the opportunity for further in-depth work on your graduation project.

IB940W - Capita selecta

Type: Standard product **Language:** Dutch **ECTS:** 1 **Price:** € 82.00

You can register for this course if you have completed all other components of the Bachelor's programme in Computer Science or the Bachelor's programme in Information Science, or are enrolled for them, and you want to end up with exactly 180 credits for the entire programme. An individual assignment will be formulated for you, taking into account the number of credits of this course. Typically, the assignment is closely linked to your graduation assignment. This avoids a familiarisation period and offers the opportunity for further in-depth work on your graduation project.

IB940X - Capita selecta

Type: Standard product **Language:** Dutch **ECTS:** 0.7 **Price:** € 54.74

You can register for this course if you have completed all other components of the Bachelor's programme in Computer Science or the Bachelor's programme in Information Science, or are enrolled for them, and you want to end up with exactly 180 credits for the entire programme. An individual assignment will be formulated for you, taking into account the number of credits of this course. Typically, the assignment is closely linked to your graduation assignment. This avoids a familiarisation period and offers the opportunity for further in-depth work on your graduation project.

IB940Y - Capita selecta

Type: Standard product **Language:** Dutch **ECTS:** 0.9 **Price:** € 68.48

You can register for this course if you have completed all other components of the Bachelor's programme in Computer Science or the Bachelor's programme in Information Science, or are enrolled for them, and you want to end up with exactly 180 credits for the entire programme. An individual assignment will be formulated for you, taking into account the number of credits of this course. Typically, the assignment is

closely linked to your graduation assignment. This avoids a familiarisation period and offers the opportunity for further in-depth work on your graduation project.

IB940Z - Capita selecta

Type: Standard product **Language:** Dutch **ECTS:** 3.5 **Price:** € 287.00

You can register for this course if you have completed all other components of the Bachelor's programme in Computer Science or the Bachelor's programme in Information Science, or are enrolled for them, and you want to end up with exactly 180 credits for the entire programme. An individual assignment will be formulated for you, taking into account the number of credits of this course. Typically, the assignment is closely linked to your graduation assignment. This avoids a familiarisation period and offers the opportunity for further in-depth work on your graduation project.

IB950A - Capita selecta

Type: Standard product **Language:** Dutch **ECTS:** 0.1 **Price:** € 13.52

You can register for this course if you have completed all other components of the Bachelor's programme in Computer Science or the Bachelor's programme in Information Science, or are enrolled for them, and you want to end up with exactly 180 credits for the entire programme. An individual assignment will be formulated for you, taking into account the number of credits of this course. Typically, the assignment is closely linked to your graduation assignment. This avoids a familiarisation period and offers the opportunity for further in-depth work on your graduation project.

IB950B - Capita selecta

Type: Standard product **Language:** Dutch **ECTS:** 1.4 **Price:** € 109.48

You can register for this course if you have completed all other components of the Bachelor's programme in Computer Science or the Bachelor's programme in Information Science, or are enrolled for them, and you want to end up with exactly 180 credits for the entire programme. An individual assignment will be formulated for you, taking into account the number of credits of this course. Typically, the assignment is closely linked to your graduation assignment. This avoids a familiarisation period and offers the opportunity for further in-depth work on your graduation project.

IB950C - Capita selecta

Type: Standard product **Language:** Dutch **ECTS:** 1.9 **Price:** € 150.48

You can register for this course if you have completed all other components of the Bachelor's programme in Computer Science or the Bachelor's programme in Information Science, or are enrolled for them, and you want to end up with exactly 180 credits for the entire programme. An individual assignment will be formulated for you, taking into account the number of credits of this course. Typically, the assignment is closely linked to your graduation assignment. This avoids a familiarisation period and offers the opportunity for further in-depth work on your graduation project.

IB950D - Capita selecta

Type: Standard product **Language:** Dutch **ECTS:** 0.8 **Price:** € 68.26

You can register for this course if you have completed all other components of the Bachelor's programme in Computer Science or the Bachelor's programme in Information Science, or are enrolled for them, and you want to end up with exactly 180 credits for the entire programme. An individual assignment will be formulated for you, taking into account the number of credits of this course. Typically, the assignment is closely linked to your graduation assignment. This avoids a familiarisation period and offers the opportunity for further in-depth work on your graduation project.

IB950E - Capita selecta

Type: Standard product **Language:** Dutch **ECTS:** 1.8 **Price:** € 150.26

You can register for this course if you have completed all other components of the Bachelor's programme in Computer Science or the Bachelor's programme in Information Science, or are enrolled for them, and you want to end up with exactly 180 credits for the entire programme. An individual assignment will be formulated for you, taking into account the number of credits of this course. Typically, the assignment is closely linked to your graduation assignment. This avoids a familiarisation period and offers the opportunity for further in-depth work on your graduation project.

IB950F - Capita selecta

Type: Standard product **Language:** Dutch **ECTS:** 4.1 **Price:** € 341.52

You can register for this course if you have completed all other components of the Bachelor's programme in Computer Science or the Bachelor's programme in Information Science, or are enrolled for them, and you want to end up with exactly 180 credits for the entire programme. An individual assignment will be formulated for you, taking into account the number of credits of this course. Typically, the assignment is closely linked to your graduation assignment. This avoids a familiarisation period and offers the opportunity for further in-depth work on your graduation project.

IB950G - Capita selecta

Type: Standard product **Language:** Dutch **ECTS:** 2.4 **Price:** € 191.48

You can register for this course if you have completed all other components of the Bachelor's programme in Computer Science or the Bachelor's programme in Information Science, or are enrolled for them, and you want to end up with exactly 180 credits for the entire programme. An individual assignment will be formulated for you, taking into account the number of credits of this course. Typically, the assignment is closely linked to your graduation assignment. This avoids a familiarisation period and offers the opportunity for further in-depth work on your graduation project.

IB950H - Capita selecta

Type: Standard product **Language:** Dutch **ECTS:** 0.4 **Price:** € 27.48

You can register for this course if you have completed all other components of the Bachelor's programme in Computer Science or the Bachelor's programme in Information Science, or are enrolled for them, and you want to end up with exactly 180 credits for the entire programme. An individual assignment will be formulated for you, taking into account the number of credits of this course. Typically, the assignment is

closely linked to your graduation assignment. This avoids a familiarisation period and offers the opportunity for further in-depth work on your graduation project.

IB950I - Capita selecta

Type: Standard product **Language:** Dutch **ECTS:** 4.2 **Price:** € 341.74

You can register for this course if you have completed all other components of the Bachelor's programme in Computer Science or the Bachelor's programme in Information Science, or are enrolled for them, and you want to end up with exactly 180 credits for the entire programme. An individual assignment will be formulated for you, taking into account the number of credits of this course. Typically, the assignment is closely linked to your graduation assignment. This avoids a familiarisation period and offers the opportunity for further in-depth work on your graduation project.

IB950J - Capita selecta

Type: Standard product **Language:** Dutch **ECTS:** 2 **Price:** € 164.00

You can register for this course if you have completed all other components of the Bachelor's programme in Computer Science or the Bachelor's programme in Information Science, or are enrolled for them, and you want to end up with exactly 180 credits for the entire programme. An individual assignment will be formulated for you, taking into account the number of credits of this course. Typically, the assignment is closely linked to your graduation assignment. This avoids a familiarisation period and offers the opportunity for further in-depth work on your graduation project.

IB950K - Capita selecta

Type: Standard product **Language:** Dutch **ECTS:** 2.3 **Price:** € 191.26

You can register for this course if you have completed all other components of the Bachelor's programme in Computer Science or the Bachelor's programme in Information Science, or are enrolled for them, and you want to end up with exactly 180 credits for the entire programme. An individual assignment will be formulated for you, taking into account the number of credits of this course. Typically, the assignment is closely linked to your graduation assignment. This avoids a familiarisation period and offers the opportunity for further in-depth work on your graduation project.

IB950L - Capita selecta

Type: Standard product **Language:** Dutch **ECTS:** 3.1 **Price:** € 259.52

You can register for this course if you have completed all other components of the Bachelor's programme in Computer Science or the Bachelor's programme in Information Science, or are enrolled for them, and you want to end up with exactly 180 credits for the entire programme. An individual assignment will be formulated for you, taking into account the number of credits of this course. Typically, the assignment is closely linked to your graduation assignment. This avoids a familiarisation period and offers the opportunity for further in-depth work on your graduation project.

IB950M - Capita selecta

Type: Standard product **Language:** Dutch **ECTS:** 3 **Price:** € 246.00

You can register for this course if you have completed all other components of the Bachelor's programme in Computer Science or the Bachelor's programme in Information Science, or are enrolled for them, and you want to end up with exactly 180 credits for the entire programme. An individual assignment will be formulated for you, taking into account the number of credits of this course. Typically, the assignment is closely linked to your graduation assignment. This avoids a familiarisation period and offers the opportunity for further in-depth work on your graduation project.

IB950N - Capita selecta

Type: Standard product **Language:** Dutch **ECTS:** 1.2 **Price:** € 95.74

You can register for this course if you have completed all other components of the Bachelor's programme in Computer Science or the Bachelor's programme in Information Science, or are enrolled for them, and you want to end up with exactly 180 credits for the entire programme. An individual assignment will be formulated for you, taking into account the number of credits of this course. Typically, the assignment is closely linked to your graduation assignment. This avoids a familiarisation period and offers the opportunity for further in-depth work on your graduation project.

IB950O - Capita selecta

Type: Standard product **Language:** Dutch **ECTS:** 3.4 **Price:** € 273.48

You can register for this course if you have completed all other components of the Bachelor's programme in Computer Science or the Bachelor's programme in Information Science, or are enrolled for them, and you want to end up with exactly 180 credits for the entire programme. An individual assignment will be formulated for you, taking into account the number of credits of this course. Typically, the assignment is closely linked to your graduation assignment. This avoids a familiarisation period and offers the opportunity for further in-depth work on your graduation project.

IB950P - Capita selecta

Type: Standard product **Language:** Dutch **ECTS:** 3.2 **Price:** € 259.74

You can register for this course if you have completed all other components of the Bachelor's programme in Computer Science or the Bachelor's programme in Information Science, or are enrolled for them, and you want to end up with exactly 180 credits for the entire programme. An individual assignment will be formulated for you, taking into account the number of credits of this course. Typically, the assignment is closely linked to your graduation assignment. This avoids a familiarisation period and offers the opportunity for further in-depth work on your graduation project.

IB950Q - Capita selecta

Type: Standard product **Language:** Dutch **ECTS:** 2.5 **Price:** € 205.00

You can register for this course if you have completed all other components of the Bachelor's programme in Computer Science or the Bachelor's programme in Information Science, or are enrolled for them, and you want to end up with exactly 180 credits for the entire programme. An individual assignment will be formulated for you, taking into account the number of credits of this course. Typically, the assignment is

closely linked to your graduation assignment. This avoids a familiarisation period and offers the opportunity for further in-depth work on your graduation project.

IB950R - Capita selecta

Type: Standard product **Language:** Dutch **ECTS:** 2.8 **Price:** € 232.26

You can register for this course if you have completed all other components of the Bachelor's programme in Computer Science or the Bachelor's programme in Information Science, or are enrolled for them, and you want to end up with exactly 180 credits for the entire programme. An individual assignment will be formulated for you, taking into account the number of credits of this course. Typically, the assignment is closely linked to your graduation assignment. This avoids a familiarisation period and offers the opportunity for further in-depth work on your graduation project.

IB950S - Capita selecta

Type: Standard product **Language:** Dutch **ECTS:** 3.3 **Price:** € 273.26

You can register for this course if you have completed all other components of the Bachelor's programme in Computer Science or the Bachelor's programme in Information Science, or are enrolled for them, and you want to end up with exactly 180 credits for the entire programme. An individual assignment will be formulated for you, taking into account the number of credits of this course. Typically, the assignment is closely linked to your graduation assignment. This avoids a familiarisation period and offers the opportunity for further in-depth work on your graduation project.

IB950T - Capita selecta

Type: Standard product **Language:** Dutch **ECTS:** 3.7 **Price:** € 300.74

You can register for this course if you have completed all other components of the Bachelor's programme in Computer Science or the Bachelor's programme in Information Science, or are enrolled for them, and you want to end up with exactly 180 credits for the entire programme. An individual assignment will be formulated for you, taking into account the number of credits of this course. Typically, the assignment is closely linked to your graduation assignment. This avoids a familiarisation period and offers the opportunity for further in-depth work on your graduation project.

IB950U - Capita selecta Bachelor's Computer Science

Dutch name: Capita selecta Bachelor Informatica **Type:** Standard product **Language:** Dutch **ECTS:** 1.7 **Price:** € 136.74

You can register for this course if you have completed all other components of the Bachelor's programme in Computer Science or the Bachelor's programme in Information Science, or are enrolled for them, and you want to end up with exactly 180 credits for the entire programme. An individual assignment will be formulated for you, taking into account the number of credits of this course. Typically, the assignment is closely linked to your graduation assignment. This avoids a familiarisation period and offers the opportunity for further in-depth work on your graduation project.

IB950V - Capita selecta Bachelor's Computer Science

Dutch name: Capita selecta Bachelor Informatica **Type:** Standard product **Language:** Dutch **ECTS:** 4.6
Price: € 382.52

You can register for this course if you have completed all other components of the Bachelor's programme in Computer Science or the Bachelor's programme in Information Science, or are enrolled for them, and you want to end up with exactly 180 credits for the entire programme. An individual assignment will be formulated for you, taking into account the number of credits of this course. Typically, the assignment is closely linked to your graduation assignment. This avoids a familiarisation period and offers the opportunity for further in-depth work on your graduation project.

IB950W - Capita selecta

Type: Standard product **Language:** Dutch **ECTS:** 0.5 **Price:** € 41.00

You can register for this course if you have completed all other components of the Bachelor's programme in Computer Science or the Bachelor's programme in Information Science, or are enrolled for them, and you want to end up with exactly 180 credits for the entire programme. An individual assignment will be formulated for you, taking into account the number of credits of this course. Typically, the assignment is closely linked to your graduation assignment. This avoids a familiarisation period and offers the opportunity for further in-depth work on your graduation project.

IB950X - Capita selecta

Type: Standard product **Language:** Dutch **ECTS:** 0.2 **Price:** € 13.74

You can register for this course if you have completed all other components of the Bachelor's programme in Computer Science or the Bachelor's programme in Information Science, or are enrolled for them, and you want to end up with exactly 180 credits for the entire programme. An individual assignment will be formulated for you, taking into account the number of credits of this course. Typically, the assignment is closely linked to your graduation assignment. This avoids a familiarisation period and offers the opportunity for further in-depth work on your graduation project.

IB950Y - Capita selecta

Type: Standard product **Language:** Dutch **ECTS:** 1.6 **Price:** € 136.52

You can register for this course if you have completed all other components of the Bachelor's programme in Computer Science or the Bachelor's programme in Information Science, or are enrolled for them, and you want to end up with exactly 180 credits for the entire programme. An individual assignment will be formulated for you, taking into account the number of credits of this course. Typically, the assignment is closely linked to your graduation assignment. This avoids a familiarisation period and offers the opportunity for further in-depth work on your graduation project.

IB950Z - Capita selecta

Type: Standard product **Language:** Dutch **ECTS:** 3.8 **Price:** € 314.26

You can register for this course if you have completed all other components of the Bachelor's programme in Computer Science or the Bachelor's programme in Information Science, or are enrolled for them, and you want to end up with exactly 180 credits for the entire programme. An individual assignment will be formulated for you, taking into account the number of credits of this course. Typically, the assignment is

closely linked to your graduation assignment. This avoids a familiarisation period and offers the opportunity for further in-depth work on your graduation project.

IB9902 - Graduation preparation

Dutch name: Voorbereiden afstuderen **Type:** Standard product **Language:** Dutch **ECTS:** 5 **Price:** € 410.00

The Bachelor's graduation in computer science is a demonstration of competence: you show that you can function at the level of an academic Bachelor's in computer science. At the same time, it is a project in which you can learn a lot, by delving into new subjects and working together with a researcher, supervisor and any team-mates. You throw yourself into a research project in which the end goal is typically not the implementation, but carrying out research, supported by software yet to be developed. Researchers affiliated with the Faculty of Science submit project proposals. As a student, you can look at these proposals before the kick-off meeting. During the kick-off meeting, the students form teams of 2 to 3 people and choose a project as a team. In exceptional cases, you can also work on a project individually. During the Graduation Preparation (VAF), you delve into the research context, carry out a literature study, draw up research questions, draw up requirements for the application you are going to develop and make an action plan for the execution during the Graduation Assignment (AF). It is compulsory to include a validation component in the action plan, in which you describe and plan how you are going to validate the results of your project in relation to the research questions. Each team member does their own sub-study. Each sub-study has its own research questions. In the project plan that is delivered at the end of the VAF, each team member has their own chapter in which their research is described.

Learning objectives

The concrete learning objectives are:

- applying subject-specific and methodological knowledge;
- integrating knowledge from different (sub)fields;
- developing solutions to formulated problems and reporting on them;
- formulating your own vision;
- accounting for choices made regarding approach and result, with an eye for reliability and validity;
- working in a responsible and careful way with reliable technologies;
- analysing, interpreting and evaluating the results found and the assignment carried out;
- evaluating the contribution of the assignment carried out to the development of the field from which the research question arose;
- formulating conclusions regarding the results found;
- formulating recommendations for further research;
- reporting in writing and orally on the results of the assignment;
- working under the conditions of the Netherlands Code of Conduct for Research Integrity;
- working together as a team.

IB9906 - Graduation project

Dutch name: Afstudeerproject **Type:** Standard product **Language:** Dutch **ECTS:** 15 **Price:** € 1,230.00

In this course, you carry out, as a team, a concluding computer science project for the Bachelor's programme. Because your project is part of a research project of a computer science researcher, you contribute (in)directly to this research. You become familiar with the research context of the project and your results contribute to the results of the research. In the project, you deliver, as a team but also personally, an integral proof of craftsmanship that can be assessed for quality. To a significant extent, the project is an integration of

previously acquired (academic) knowledge and skills, and you apply methods and techniques. The project is complex enough to test the Bachelor's level. The project is subject-specific and broad enough to justify the degree (Bachelor's in Computer Science). The assessment is instrumented in such a way that an acceptable minimum academic Bachelor's level is guaranteed on passing. You report on your project result, its place in the research and its contribution to the research in a thesis. In a personal chapter, you describe, in an academically sound way, the result of an in-depth study of an important aspect of the project. In such a project, your aim is to successfully complete your Bachelor's curriculum. All 'compulsory' subject-specific matters have been addressed. In the Bachelor's graduation project in computer science, you learn to integrate the various competences (academic and subject-specific competences). To that end, you go through a process that resembles sufficiently what professional practice will ask. This takes place within a protected learning situation, in which you are supervised by a teacher/researcher. In the project, you are enabled to show your academic working and thinking level. Although this explicitly cannot be a 'masterpiece', it does have the characteristics of one; you can show the 'world' that you have reached the level of a Bachelor's in Computer Science. As a student, you therefore have something in hand, or in your portfolio, with which a current and/or future employer, but also a teacher in a follow-up programme, can assess the meaning and quality of the degree.

IM0004 - Computer Science graduation assignment preparation

Dutch name: Voorbereiden afstudeeropdracht CS **Type:** Standard product **Language:** Dutch **ECTS:** 10
Price: € 820.00

During this module, you show that, through literature study and possibly preliminary research, you can, for a chosen assignment:

- formulate it as a research assignment;
- place it in a context;
- provide it with a suitable intended approach and solution method;
- divide it into phases and describe and plan these phases;
- account for the choices made.

You start with an orientation. You read the graduation guide: it is made available digitally via the digital learning environment. You can also consult the digital learning environment for information on subjects, graduation supervisors, the graduation process and examples of graduation assignments. You can also attend graduation meetings, graduation presentations or the research meetings. The next step is to make contact with a graduation supervisor. After an orientation conversation, you make your choice known by completing the online registration form Graduation track CS. You are assigned a graduation committee and receive instructions to register for IM0004. During the preparation of your graduation assignment, it may become clear that studying additional material is necessary. In consultation with your graduation committee, it is determined which in-depth study in a particular computer science direction, or which specific domain knowledge of the domain within which your research takes place, is necessary. You use this in-depth study when writing your research plan. After your research plan has been approved and assessed by your graduation committee, you can continue the graduation track with IM990C Computer Science Graduation Assignment.

IM000S - CS Graduation Assignment Preparation

Type: Standard product **Language:** Dutch **ECTS:** 5 **Price:** € 410.00

During this module, you show that, through literature study and possibly preliminary research, you can, for a chosen assignment:

- formulate it as a research assignment;
- place it in a context;
- provide it with a suitable intended approach and solution method;
- divide it into phases and describe and plan these phases;
- account for the choices made.

You start the preparation of the graduation track with an orientation. You read the graduation guide: it is made available digitally via the digital learning environment. You can also consult the digital learning environment for information on subjects, graduation supervisors, the graduation process and examples of graduation assignments. You can also attend graduation meetings and graduation presentations. The next step is to make contact with a graduation supervisor. After an orientation conversation, you make your choice known by completing the online registration form Graduation track CS. You are assigned a graduation committee and receive instructions to register for CS Graduation Assignment Preparation. During the preparation of your graduation assignment, it may become clear that studying additional material is necessary. In consultation with your graduation committee, it is determined which in-depth study in a particular computer science direction, or which specific domain knowledge of the domain within which your research takes place, is necessary. You use this in-depth study when writing your research plan. After your research plan has been approved and assessed by your graduation committee, you can continue the graduation track with IM990C Computer Science Graduation Assignment.

IM000T - CS Graduation Assignment Preparation

Type: Standard product **Language:** Dutch **ECTS:** 7.5 **Price:** € 615.00

During this module, you show that, through literature study and possibly preliminary research, you can, for a chosen assignment:

- formulate it as a research assignment;
- place it in a context;
- provide it with a suitable intended approach and solution method;
- divide it into phases and describe and plan these phases;
- account for the choices made.

You start the preparation of the graduation track with an orientation. You read the graduation guide: it is made available digitally via the digital learning environment. You can also consult the digital learning environment for information on subjects, graduation supervisors, the graduation process and examples of graduation assignments. You can also attend graduation meetings and graduation presentations. The next step is to make contact with a graduation supervisor. After an orientation conversation, you make your choice known by completing the online registration form Graduation track CS. You are assigned a graduation committee and receive instructions to register for CS Graduation Assignment Preparation. During the preparation of your graduation assignment, it may become clear that studying additional material is necessary. In consultation with your graduation committee, it is determined which in-depth study in a particular computer science direction, or which specific domain knowledge of the domain within which your research takes place, is necessary. You use this in-depth study when writing your research plan. After your research plan has been approved and assessed by your graduation committee, you can continue the graduation track with IM990C Computer Science Graduation Assignment.

IM000V - Computer Science graduation assignment preparation

Dutch name: Voorbereiden afstudeeropdracht CS **Type:** Standard product **Language:** Dutch **ECTS:** 11
Price: € 902.00

During this module, you show that, through literature study and possibly preliminary research, you can, for a chosen assignment:

- formulate it as a research assignment;
- place it in a context;
- provide it with a suitable intended approach and solution method;
- divide it into phases and describe and plan these phases;
- account for the choices made.

You start the preparation of the graduation track with an orientation. You read the graduation guide: it is made available digitally via the digital learning environment. You can also consult the digital learning environment for information on subjects, graduation supervisors, the graduation process and examples of graduation assignments. You can also attend graduation meetings and graduation presentations. The next step is to make contact with a graduation supervisor. After an orientation conversation, you make your choice known by completing the online registration form Graduation track CS. You are assigned a graduation committee and receive instructions to register for CS Graduation Assignment Preparation. During the preparation of your graduation assignment, it may become clear that studying additional material is necessary. In consultation with your graduation committee, it is determined which in-depth study in a particular computer science direction, or which specific domain knowledge of the domain within which your research takes place, is necessary. You use this in-depth study when writing your research plan. After your research plan has been approved and assessed by your graduation committee, you can continue the graduation track with IM990C Computer Science Graduation Assignment.

IM000W - Computer Science graduation assignment preparation

Dutch name: Voorbereiden afstudeeropdracht CS **Type:** Standard product **Language:** Dutch **ECTS:** 12
Price: € 984.00

During this module, you show that, through literature study and possibly preliminary research, you can, for a chosen assignment:

- formulate it as a research assignment;
- place it in a context;
- provide it with a suitable intended approach and solution method;
- divide it into phases and describe and plan these phases;
- account for the choices made.

You start the preparation of the graduation track with an orientation. You read the graduation guide: it is made available digitally via the digital learning environment. You can also consult the digital learning environment for information on subjects, graduation supervisors, the graduation process and examples of graduation assignments. You can also attend graduation meetings and graduation presentations. The next step is to make contact with a graduation supervisor. After an orientation conversation, you make your choice known by completing the online registration form Graduation track CS. You are assigned a graduation committee and receive instructions to register for CS Graduation Assignment Preparation. During the preparation of your graduation assignment, it may become clear that studying additional material is necessary. In consultation with your graduation committee, it is determined which in-depth study in a particular computer science direction, or which specific domain knowledge of the domain within which your research takes place, is necessary. You use this in-depth study when writing your research plan. After your

research plan has been approved and assessed by your graduation committee, you can continue the graduation track with IM990C Computer Science Graduation Assignment.

IM000X - Computer Science graduation assignment preparation

Dutch name: Voorbereiden afstudeeropdracht CS **Type:** Standard product **Language:** Dutch **ECTS:** 13
Price: € 1,066.00

During this module, you show that, through literature study and possibly preliminary research, you can, for a chosen assignment:

- formulate it as a research assignment;
- place it in a context;
- provide it with a suitable intended approach and solution method;
- divide it into phases and describe and plan these phases;
- account for the choices made.

You start the preparation of the graduation track with an orientation. You read the graduation guide: it is made available digitally via the digital learning environment. You can also consult the digital learning environment for information on subjects, graduation supervisors, the graduation process and examples of graduation assignments. You can also attend graduation meetings and graduation presentations. The next step is to make contact with a graduation supervisor. After an orientation conversation, you make your choice known by completing the online registration form Graduation track CS. You are assigned a graduation committee and receive instructions to register for CS Graduation Assignment Preparation. During the preparation of your graduation assignment, it may become clear that studying additional material is necessary. In consultation with your graduation committee, it is determined which in-depth study in a particular computer science direction, or which specific domain knowledge of the domain within which your research takes place, is necessary. You use this in-depth study when writing your research plan. After your research plan has been approved and assessed by your graduation committee, you can continue the graduation track with IM990C Computer Science Graduation Assignment.

IM000Y - Computer Science graduation assignment preparation

Dutch name: Voorbereiden afstudeeropdracht CS **Type:** Standard product **Language:** Dutch **ECTS:** 14
Price: € 1,148.00

During this module, you show that, through literature study and possibly preliminary research, you can, for a chosen assignment:

- formulate it as a research assignment;
- place it in a context;
- provide it with a suitable intended approach and solution method;
- divide it into phases and describe and plan these phases;
- account for the choices made.

You start the preparation of the graduation track with an orientation. You read the graduation guide: it is made available digitally via the digital learning environment. You can also consult the digital learning environment for information on subjects, graduation supervisors, the graduation process and examples of graduation assignments. You can also attend graduation meetings and graduation presentations. The next step is to make contact with a graduation supervisor. After an orientation conversation, you make your choice known by completing the online registration form Graduation track CS. You are assigned a graduation committee and receive instructions to register for CS Graduation Assignment Preparation. During the preparation of your graduation assignment, it may become clear that studying additional material is

necessary. In consultation with your graduation committee, it is determined which in-depth study in a particular computer science direction, or which specific domain knowledge of the domain within which your research takes place, is necessary. You use this in-depth study when writing your research plan. After your research plan has been approved and assessed by your graduation committee, you can continue the graduation track with IM990C Computer Science Graduation Assignment.

IM000Z - Computer Science graduation assignment preparation

Dutch name: Voorbereiden afstudeeropdracht CS **Type:** Standard product **Language:** Dutch **ECTS:** 15
Price: € 1,230.00

During this module, you show that, through literature study and possibly preliminary research, you can, for a chosen assignment:

- formulate it as a research assignment;
- place it in a context;
- provide it with a suitable intended approach and solution method;
- divide it into phases and describe and plan these phases;
- account for the choices made.

You start the preparation of the graduation track with an orientation. You read the graduation guide: it is made available digitally via the digital learning environment. You can also consult the digital learning environment for information on subjects, graduation supervisors, the graduation process and examples of graduation assignments. You can also attend graduation meetings and graduation presentations. The next step is to make contact with a graduation supervisor. After an orientation conversation, you make your choice known by completing the online registration form Graduation track CS. You are assigned a graduation committee and receive instructions to register for CS Graduation Assignment Preparation. During the preparation of your graduation assignment, it may become clear that studying additional material is necessary. In consultation with your graduation committee, it is determined which in-depth study in a particular computer science direction, or which specific domain knowledge of the domain within which your research takes place, is necessary. You use this in-depth study when writing your research plan. After your research plan has been approved and assessed by your graduation committee, you can continue the graduation track with IM990C Computer Science Graduation Assignment.

IM0203 - Software Architecture

Type: Standard product **Language:** Dutch **ECTS:** 7.5 **Price:** € 594.00

The structure of software systems is often extremely complex: such a system can consist of millions of lines of program code, built up of several components that have to cooperate in different ways and operate in a distributed setting. Stringent requirements imposed on this kind of system, such as a short time-to-market, extreme reliability and robustness, or hard real-time constraints, are challenges that must be taken into account from the very beginning. Software architecture is the field that deals with describing and analysing a system with the help of models to meet the (often conflicting) set of requirements drawn up on the basis of the concerns and interests of the stakeholders. An estimation error in the initial design can have far-reaching and disastrous consequences for the success of a software project. The course Software Architecture consists of two parts: a theoretical and a practical part. The theoretical part is divided into twelve learning units. The subjects are discussed on the basis of the workbook, a reader with articles, and a textbook. A broad overview is given of what software architecture is, with an emphasis on models and techniques that can be used in designing. In the practical part, you are asked to apply the theory and design an architecture yourself for an existing software system. A software architecture is a high-level description of the components of which a system is built up and how these components cooperate with each other. The central theme of this course

consists of taking stock of the interests of the various parties involved (the stakeholders), of the requirements for the system that can be derived from them, and of describing and evaluating a software architecture. Different parties often have conflicting interests, and these will have to be weighed against each other as well as possible by the software architect. Formulating and categorising requirements unambiguously is done on the basis of a quality model. Various techniques for describing an architecture are also dealt with, such as Kruchten's 4+1 model, the viewpoints of Rozanski and Woods, and the formal Architectural Description Languages (ADL). On the basis of scenarios and use cases, an architecture can then be evaluated against the requirements drawn up. This course also addresses the characteristics of business applications, such as software for bookkeeping or for scheduling tasks. These applications often have a similar architecture and this can be reused. It is also discussed how these business applications can be further integrated within a company. Other subjects covered include, for example, the REST architectural [...]

IM0502 - Software Engineering graduation assignment preparation

Dutch name: Voorbereiden afstudeeropdracht SE **Type:** Standard product **Language:** Dutch **ECTS:** 5
Price: € 410.00

Learning objectives: during this module, you show that, through literature study and possibly preliminary research, you can, for a chosen assignment:

- formulate it as a research assignment;
- place it in a context;
- provide it with a suitable intended approach and solution method;
- divide it into phases and describe and plan these phases;
- account for the choices made.

You start with an orientation. You read the graduation guide: it is made available digitally via the online learning environment. You can also consult the online learning environment for information on subjects, graduation supervisors, the graduation process and examples of graduation assignments. You can also attend graduation meetings, graduation presentations or the research meetings. The next step is to make contact with a graduation supervisor. After an orientation conversation, you make your choice known by completing the online registration form Graduation track SE. You are assigned a graduation committee and receive instructions to register for IM050V. During the preparation of your graduation assignment, it may become clear that studying additional material is necessary. In consultation with your graduation committee, it is determined which in-depth study in a particular computer science direction, or which specific domain knowledge of the domain within which your research takes place, is necessary. You use this in-depth study when writing your research plan. After your research plan has been approved and assessed by your graduation committee, you can continue the graduation track with IM9906 Software Engineering Graduation Assignment.

IM050V - Software Engineering graduation assignment preparation

Dutch name: Voorbereiden afstudeeropdracht SE **Type:** Standard product **Language:** Dutch **ECTS:** 6
Price: € 492.00

Learning objectives: during this module, you show that, through literature study and possibly preliminary research, you can, for a chosen assignment:

- formulate it as a research assignment;
- place it in a context;
- provide it with a suitable intended approach and solution method;
- divide it into phases and describe and plan these phases;

- account for the choices made.

You start the preparation of the graduation track with an orientation. You read the graduation guide: it is made available digitally via the digital learning environment. You can also consult the digital learning environment for information on subjects, graduation supervisors, the graduation process and examples of graduation assignments. You can also attend graduation meetings and graduation presentations. The next step is to make contact with a graduation supervisor. After an orientation conversation, you make your choice known by completing the online registration form Graduation track SE. You are assigned a graduation committee and receive instructions to register for SE Graduation Assignment Preparation. During the preparation of your graduation assignment, it may become clear that studying additional material is necessary. In consultation with your graduation committee, it is determined which in-depth study in a particular computer science direction, or which specific domain knowledge of the domain within which your research takes place, is necessary. You use this in-depth study when writing your research plan. After your research plan has been approved and assessed by your graduation committee, you can continue the graduation track with IM9906 Software Engineering Graduation Assignment.

IM050W - Software Engineering graduation assignment preparation

Dutch name: Voorbereiden afstudeeropdracht SE **Type:** Standard product **Language:** Dutch **ECTS:** 7
Price: € 574.00

Learning objectives: during this module, you show that, through literature study and possibly preliminary research, you can, for a chosen assignment:

- formulate it as a research assignment;
- place it in a context;
- provide it with a suitable intended approach and solution method;
- divide it into phases and describe and plan these phases;
- account for the choices made.

You start the preparation of the graduation track with an orientation. You read the graduation guide: it is made available digitally via the digital learning environment. You can also consult the digital learning environment for information on subjects, graduation supervisors, the graduation process and examples of graduation assignments. You can also attend graduation meetings and graduation presentations. The next step is to make contact with a graduation supervisor. After an orientation conversation, you make your choice known by completing the online registration form Graduation track SE. You are assigned a graduation committee and receive instructions to register for SE Graduation Assignment Preparation. During the preparation of your graduation assignment, it may become clear that studying additional material is necessary. In consultation with your graduation committee, it is determined which in-depth study in a particular computer science direction, or which specific domain knowledge of the domain within which your research takes place, is necessary. You use this in-depth study when writing your research plan. After your research plan has been approved and assessed by your graduation committee, you can continue the graduation track with IM9906 Software Engineering Graduation Assignment.

IM050X - Software Engineering graduation assignment preparation

Dutch name: Voorbereiden afstudeeropdracht SE **Type:** Standard product **Language:** Dutch **ECTS:** 8
Price: € 656.00

Learning objectives: during this module, you show that, through literature study and possibly preliminary research, you can, for a chosen assignment:

- formulate it as a research assignment;

- place it in a context;
- provide it with a suitable intended approach and solution method;
- divide it into phases and describe and plan these phases;
- account for the choices made.

You start the preparation of the graduation track with an orientation. You read the graduation guide: it is made available digitally via the digital learning environment. You can also consult the digital learning environment for information on subjects, graduation supervisors, the graduation process and examples of graduation assignments. You can also attend graduation meetings and graduation presentations. The next step is to make contact with a graduation supervisor. After an orientation conversation, you make your choice known by completing the online registration form Graduation track SE. You are assigned a graduation committee and receive instructions to register for SE Graduation Assignment Preparation. During the preparation of your graduation assignment, it may become clear that studying additional material is necessary. In consultation with your graduation committee, it is determined which in-depth study in a particular computer science direction, or which specific domain knowledge of the domain within which your research takes place, is necessary. You use this in-depth study when writing your research plan. After your research plan has been approved and assessed by your graduation committee, you can continue the graduation track with IM9906 Software Engineering Graduation Assignment.

IM050Y - Software Engineering graduation assignment preparation

Dutch name: Voorbereiden afstudeeropdracht SE **Type:** Standard product **Language:** Dutch **ECTS:** 9
Price: € 738.00

Learning objectives: during this module, you show that, through literature study and possibly preliminary research, you can, for a chosen assignment:

- formulate it as a research assignment;
- place it in a context;
- provide it with a suitable intended approach and solution method;
- divide it into phases and describe and plan these phases;
- account for the choices made.

You start the preparation of the graduation track with an orientation. You read the graduation guide: it is made available digitally via the digital learning environment. You can also consult the digital learning environment for information on subjects, graduation supervisors, the graduation process and examples of graduation assignments. You can also attend graduation meetings and graduation presentations. The next step is to make contact with a graduation supervisor. After an orientation conversation, you make your choice known by completing the online registration form Graduation track SE. You are assigned a graduation committee and receive instructions to register for SE Graduation Assignment Preparation. During the preparation of your graduation assignment, it may become clear that studying additional material is necessary. In consultation with your graduation committee, it is determined which in-depth study in a particular computer science direction, or which specific domain knowledge of the domain within which your research takes place, is necessary. You use this in-depth study when writing your research plan. After your research plan has been approved and assessed by your graduation committee, you can continue the graduation track with IM9906 Software Engineering Graduation Assignment.

IM050Z - Software Engineering graduation assignment preparation

Dutch name: Voorbereiden afstudeeropdracht SE **Type:** Standard product **Language:** Dutch **ECTS:** 10 **Price:** € 820.00

Learning objectives: during this module, you show that, through literature study and possibly preliminary research, you can, for a chosen assignment:

- formulate it as a research assignment;
- place it in a context;
- provide it with a suitable intended approach and solution method;
- divide it into phases and describe and plan these phases;
- account for the choices made.

You start the preparation of the graduation track with an orientation. You read the graduation guide: it is made available digitally via the digital learning environment. You can also consult the digital learning environment for information on subjects, graduation supervisors, the graduation process and examples of graduation assignments. You can also attend graduation meetings and graduation presentations. The next step is to make contact with a graduation supervisor. After an orientation conversation, you make your choice known by completing the online registration form Graduation track SE. You are assigned a graduation committee and receive instructions to register for SE Graduation Assignment Preparation. During the preparation of your graduation assignment, it may become clear that studying additional material is necessary. In consultation with your graduation committee, it is determined which in-depth study in a particular computer science direction, or which specific domain knowledge of the domain within which your research takes place, is necessary. You use this in-depth study when writing your research plan. After your research plan has been approved and assessed by your graduation committee, you can continue the graduation track with IM9906 Software Engineering Graduation Assignment.

IM0603 - Design for change

Type: Standard product **Language:** Dutch **ECTS:** 7.5 **Price:** € 594.00

Software is almost never ‘finished’. The requirements for software change constantly: again and again, new features have to be added, or existing functionality has to be changed. One of the most important requirements for software is therefore that it must be easy to adapt, in other words to change. The course consists of two parts. In the first part, you add a new function to existing software and make it more flexible for future changes. For the first part, we use a simplified version of domain-driven design (DDD). That is a first step in optimising software with a view to future changes. Using an extensive case, we show you step by step how you can analyse a domain and how you use that analysis when drawing up a design. In that design, you will often use design patterns to build in flexibility. Design patterns form the second building block in optimising software with a view to flexibility. In addition, in this course you learn to write an academic text. You base your article on the assignment you have carried out. You use that assignment as a case study. The emphasis is on writing a good abstract, introduction, research methods, and a list of academic sources to support your article.

Learning objectives

After studying this course, you can:

- carry out a domain analysis based on a problem description, possibly supplemented with external sources;
- draw up a design in which each entity (method, class, package) has a single, clear responsibility;
- separate the creation and use of objects in a design;
- understand and apply design principles in a design;

- describe a design with the help of diagrams and text, with an emphasis on the responsibilities;
- draw up a design that is optimised for future changes;
- argue how design decisions and design patterns contribute to the flexibility of a design;
- use design patterns to optimise a design for future changes;
- write a motivating and clear introduction for an academic article;
- define the research setting, including research questions, research strategy and data collection and analysis processes for an academic article;
- find relevant academic sources;
- refer to academic sources in the right way;
- write an abstract for an academic article.

IM0703 - Software quality management

Type: Standard product **Language:** Dutch **ECTS:** 7.5 **Price:** € 615.00

Software is essential in our world. It is therefore important to guarantee high software quality during its development and operation. This course addresses the processes around developing and maintaining software. On the one hand, an overview is given of the aspects that are important for successfully carrying out software development projects. Various models have been developed for this over time. These describe the order and the way in which a number of activities should be carried out in order to arrive at a successful software product. Each model has its own starting points and, when applied in practice, specific advantages and disadvantages. Many models are in use, although in almost every situation there is a mix of one or more models. Knowledge of important aspects of process models is therefore important in order to be able to choose a suitable model for a concrete product in practice. We look at the models themselves, but also go further into the processes that operate within and outside the software process. On the other hand, the general theory of the subjects of software maintenance and evolution, software renovation, program analysis and program transformation is addressed. The emphasis here is on the first subject. Brief attention is also paid to mining of software repositories. The course has an essay assignment and a practicum assignment. In the essay assignment, you are asked to pose research questions in the area of software development processes and to write these up in a well-reasoned essay. You must also make peer reviews of the draft version of other students' essays. On the basis of the reviews you receive from your fellow students, you can finish the final essay. In this assignment, you are expected to be able to combine the theory and practice of software process models, to use the relevant literature and to express your own opinion in writing at an academic level. In the practicum assignment, you are asked to work out a number of software metrics and to test them on existing systems of different sizes, written in Java. The metrics used come from a quality model and give insight into the structure and quality of a system. Conclusions must then be drawn from the results found about the maintainability of the software and the risk areas. This assignment uses the domain-specific programming language Rascal, which is intended for analysing and transforming program code. Rascal is basically an imperative programming language, with a syntax that closely resembles that of Java. In addition, the language has a number of constructs for easily programming at meta level: there is support for analysing projects and Java files, there are special data structures for the [...]

IM0712 - Key Topics in Artificial Intelligence

Type: Standard product **Language:** Dutch **ECTS:** 5 **Price:** € 410.00

This starter course offers an introduction to a number of main areas of artificial intelligence. The course deals at a fast pace with important, technical subjects across the whole breadth of AI. A first part is about the history of ideas of AI, including Cybernetics. A second part, the main focus of the subject, deals with the ideas, concepts and techniques of a number of important directions within AI. The third part consists of a critical

analysis of future developments of AI, foreseen and unforeseen, and their consequences for society. During the course, you also work in small groups on a small assignment. In the practical assignments, an existing system or existing experiment must be used via the programming language Python for an exercise or experiment. The assignments are chosen from a list of possibilities presented in advance by the teachers. The combination of a broad, substantive introduction to the field and a hands-on exploration of the possibilities of AI systems makes this subject an effective introduction to the Master's in AI.

Learning objectives

After completing this course, you can:

- recognise and name the most important areas of AI (Knowledge-based AI, Machine Learning, Natural Language Processing, Computer Vision, Robotics, Responsible AI), describe the history and the most important directions in AI, and reflect on the place of the field of AI in our society;
- interpret technical and conceptual challenges in sub-fields of AI, giving a critical view of the (im)possibilities in those sub-fields;
- give a well-founded vision of how current developments in AI can lead to new technological breakthroughs, but also how they can lead to unforeseen consequences, in particular what the possible ethical consequences and new responsibilities are;
- on a small scale, use existing AI algorithms, or Python implementations of AI systems, to replicate theoretical explorations or practical experiments or to reuse systems in a new context;
- effectively know and search sources to acquire new knowledge about AI, such as from journals, conferences, websites and books, but also from online communities and other online sources to keep up to date in the field;
- adequately analyse an aspect of current AI, analyse it critically and report clearly in writing what the challenges are in that area, and what innovations in this sub-field can lead to.

IM0802 - Responsible Artificial Intelligence

Type: Standard product **Language:** Dutch **ECTS:** 5 **Price:** € 410.00

The responsible use of artificial intelligence techniques is very important in an era in which more and more use is made of smart algorithms, and in which this technology is ubiquitous. This course is about the ethical sides of AI. On the one hand, this concerns an ethical analysis of the consequences of different types of algorithms, such as prediction and profiling, experimentation and data linkage. On the other hand, it concerns technical solutions for ethical consequences of algorithms, for example through explainability and interpretability of algorithms, and the development of algorithms that are fairer (and what is fair then) or that explicitly avoid certain biases, or even algorithms that can themselves reason about ethical trade-offs in complex situations. The course teaches a systematic approach for studying the consequences of algorithms, and it deals with modern, technical solutions for responsible and safe AI.

Learning objectives

After studying this course, you:

- can describe various ethical viewpoints, recognise the specific characteristics of ethics in a digital world, on that basis describe the field of 'responsible AI', characterise which ethical challenges are old and new, interpret what their impact is and explain why;
- can describe the main directions in the current state of affairs regarding responsible AI, know the most important concepts and viewpoints, know the most important social and scientific developments, have explored one or several sub-field(s) in more depth, and can reflect, also together with others, on the responsible deployment of AI, from the different viewpoints of stakeholders;

- can systematically analyse and problematise a given technological development or domain regarding responsible AI, relate it to academic literature, with attention to the consequences and stakeholders, give a scientifically substantiated and systematically argued analysis of it, and report on it in writing;
- have knowledge and insight into how consequences of technological developments in AI can on the one hand be prevented by means of scientific, socio-societal or legal measures, and on the other hand be mitigated by means of design-technical and technological solutions from AI, and can describe, contrast and evaluate both types of solutions;
- can, in a team, systematically analyse an intended application of AI in a domain from a design perspective, can use appropriate tools in the form of guidelines and design principles to arrive at a better design, and can put forward solutions for specific ethical challenges of AI technology, to avoid or reduce undesirable effects, and can report on this in writing.

IM0813 - Software Security

Type: Standard product **Language:** Dutch **ECTS:** 7.5 **Price:** € 594.00

The course consists of two parts. The first part is devoted to software security, in which you acquire knowledge and skills regarding securing software. The second part is devoted to academic skills, in which you acquire knowledge and skills regarding research methods and formulating research questions.

Learning objectives

After completing the software security part of this course, you can:

- name and explain a number of common security problems in software;
- explain the underlying causes of these security problems, situated in specific applications and application types, programming languages and platforms and operating systems;
- indicate which activities (‘best practices’) in the software development process must be carried out to create secure software, including risk analysis and static code analysis;
- give principles and guidelines to prevent or detect security problems;
- name and explain language-based and platform-based solutions, such as sandboxing;
- deal responsibly with ethical aspects of software security.

After completing the academic skills part of this course, you can:

- summarise the structure of an academic article;
- recognise the research question in an academic article and examine to what extent the research carried out answers this question;
- list the limitations of a study that may lead to follow-up questions;
- indicate what the contribution of an article is relative to the existing literature;
- recognise the research method used;
- determine whether data were collected and analysed for the study, and if so, which;
- give a coherent argument about the content of an academic article;
- write down ideas and opinions clearly and substantiate them with facts;
- formulate a research question of your own as a follow-up to existing research.

IM0823 - Software security

Type: Standard product **Language:** Dutch **ECTS:** 7.5 **Price:** € 615.00

The course consists of two parts. The first part is devoted to software security, in which you acquire knowledge and skills regarding securing software. The second part is devoted to academic skills, in which you acquire knowledge and skills regarding research methods and formulating research questions.

Learning objectives

After completing the software security part of this course, you can:

- name and explain a number of common security problems in software;
- explain the underlying causes of these security problems, situated in specific applications and application types, programming languages and platforms and operating systems;
- indicate which activities ('best practices') in the software development process must be carried out to create secure software, including risk analysis and static code analysis;
- give principles and guidelines to prevent or detect security problems;
- name and explain language-based and platform-based solutions, such as sandboxing;
- deal responsibly with ethical aspects of software security.

After completing the academic skills part of this course, you can:

- summarise the structure of an academic article;
- recognise the research question in an academic article and examine to what extent the research carried out answers this question;
- list the limitations of a study that may lead to follow-up questions;
- indicate what the contribution of an article is relative to the existing literature;
- recognise the research method used;
- determine whether data were collected and analysed for the study, and if so, which;
- give a coherent argument about the content of an academic article;
- write down ideas and opinions clearly and substantiate them with facts;
- formulate a research question of your own as a follow-up to existing research.

IM0902 - Bayesian Reasoning and Learning

Type: Standard product **Language:** Dutch **ECTS:** 5 **Price:** € 410.00

Many Artificial Intelligence systems have to be able to deal with uncertainty in one way or another. This course deals with a number of the most important techniques for representing uncertainty in the form of probability distributions, reasoning with probability models, and learning these probability models from data. Common models are so-called 'probabilistic graphical models', and these include variants such as hidden Markov models, Bayesian networks, and also decision-theoretic models such as influence diagrams. These models are widely applied in other fields of science, such as medicine and biology. In addition to statistical models, this course will also deal with causal models. Many relevant questions for agents are causal in nature, for example: what is the effect of an action? In science too, the important questions are causal, for example: which side effects are caused by a medicine? In recent years, methods have been developed for reasoning with causal knowledge, and even for learning this kind of knowledge from previously collected data. The course contains a combination of theory and practical application. The theoretical part is assessed by means of a digital group exam with open questions. In addition, the course contains assignments in which Bayesian networks are applied from a more practical perspective.

Learning objectives

After completing this course, you can:

- make a reasoned choice to apply probabilistic principles for learning and reasoning under uncertainty;
- draw the right statistical and causal conclusions from a causal model;
- apply the difference between statistical and causal inference to report results from machine learning models in a responsible way;
- name algorithms and properties of algorithms for reasoning with probability distributions and learning from data, and reason about them in order to find a suitable solution to a problem statement;

- use probabilistic graphical models to represent and solve sequential decision problems;
- understand modern academic literature on probabilistic and causal models and relate it to the most important concepts in this field;
- use modern software tools to apply Bayesian networks in practice.

IM0903 - System verification and testing

Type: Standard product **Language:** Dutch **ECTS:** 7.5 **Price:** € 615.00

Verification and testing are two important aspects of the construction of correct systems. A crucial instrument in this is the use of models. Models are central to this course. The first part of this course is about drawing up models in mathematical formalisms such as Kripke structures and timed automata. Properties that specify the correct behaviour of the system can be expressed over these models. These properties lead to a formal specification. We deal with temporal logic, a mathematical language capable of expressing such properties over the time of the system. For example: an air traffic control system must satisfy the temporal property that two aircraft never collide with each other. In the second part of the course, you prove by using model checking that the models satisfy their temporal specifications. Model checking is nowadays a standard technique for designing hardware and is becoming increasingly popular in software design. In 2007, the developers of this technique received the ACM Turing Award (the unofficial Nobel Prize in computer science). The third part of the course is about testing systems. In contrast to the first part, the analysis is not about a model but about a system that can run on a machine. Testing is nevertheless based on a model of the same type as in the first part of the course. This way of testing is therefore called model-based testing. This technique is new and is becoming increasingly popular in software development by large companies. In addition, we will study an approach to testing existing software that makes it efficient to find bugs, even when no explicit model of the system was made in advance. This course gives an introduction to the theories behind model checking and testing. In all parts of the course, these theoretical concepts are applied to concrete models and software.

Learning objectives

After completing this course, you can:

- make formal models of software systems;
- write formal specifications using temporal logics;
- understand basic principles of equivalence between models;
- recognise in which situations model checking is applicable;
- use a model checker to prove that the models satisfy their formal specifications;
- name and explain the basic principles of testing techniques, such as model-based testing and scriptless testing;
- use a model-based testing tool to test whether an implementation corresponds to the specification model;
- apply a scriptless testing approach to testing existing software.

IM1002 - Machine Learning

Type: Standard product **Language:** Dutch **ECTS:** 5 **Price:** € 396.00

This course deals with various paradigms from machine learning such as supervised, unsupervised and reinforcement learning. For each of these paradigms, relevant concepts, techniques and technologies are dealt with both theoretically and practically in a data-oriented approach. Specifically, the underlying mathematical properties of learning algorithms are explained in order to understand how accurate and efficient ML models can be designed. The course further focuses on creating the academic skills needed to

apply, modify, create and optimise ML models for various socially relevant problems. Finally, the course deals with relevant ethical and social aspects of ML models, i.e. interpretability, to ensure that the decisions made in designing the models and the results of the models are understandable to others.

Learning objectives

After studying the course, you can:

- characterise the fundamental/underlying principles of ML that are seen as optimisation problems in the context of data, and know how to distinguish different subgroups such as supervised, unsupervised and reinforcement learning with the correct terminology;
- for a given problem, choose the right methodology and techniques in order to construct an empirical experimental set-up within which a solution is sought with the right choice of model design, data processing steps and ML methods;
- build a worked-out experimental set-up using available datasets, ML code libraries, and with the support of relevant academic literature;
- evaluate ML-based solutions using suitable evaluation metrics, relate results to the original problem statement, and compare solutions with each other on the basis of objective criteria;
- understand and explain the need for interpretability of ML, and have knowledge and insight of the various ways of making this need clear with techniques such as data visualisation methods, and outcomes;
- present an ML experiment in writing in a scientific way with associated reporting formulations regarding the design, the choices made, the experimental set-up, the results obtained and the conclusions, and know how to embed that process in the academic literature.

IM1012 - Machine Learning

Type: Standard product **Language:** Dutch **ECTS:** 5 **Price:** € 410.00

This course deals with various paradigms from machine learning such as supervised, unsupervised and reinforcement learning. For each of these paradigms, relevant concepts, techniques and technologies are dealt with both theoretically and practically in a data-oriented approach. Specifically, the underlying mathematical properties of learning algorithms are explained in order to understand how accurate and efficient ML models can be designed. The course further focuses on creating the academic skills needed to apply, modify, create and optimise ML models for various socially relevant problems. Finally, the course deals with relevant ethical and social aspects of ML models, i.e. interpretability, to ensure that the decisions made in designing the models and the results of the models are understandable to others.

Learning objectives

After studying the course, you can:

- characterise the fundamental/underlying principles of ML that are seen as optimisation problems in the context of data, and know how to distinguish different subgroups such as supervised, unsupervised and reinforcement learning with the correct terminology;
- for a given problem, choose the right methodology and techniques in order to construct an empirical experimental set-up within which a solution is sought with the right choice of model design, data processing steps and ML methods;
- build a worked-out experimental set-up using available datasets, ML code libraries, and with the support of relevant academic literature;
- evaluate ML-based solutions using suitable evaluation metrics, relate results to the original problem statement, and compare solutions with each other on the basis of objective criteria;

- understand and explain the need for interpretability of ML, and have knowledge and insight of the various ways of making this need clear with techniques such as data visualisation methods, and outcomes;
- present an ML experiment in writing in a scientific way with associated reporting formulations regarding the design, the choices made, the experimental set-up, the results obtained and the conclusions, and know how to embed that process in the academic literature.

IM1102 - Deep Neural Engineering

Type: Standard product **Language:** Dutch **ECTS:** 5 **Price:** € 396.00

The course focuses on neural network methods and models that deal with complex prediction and optimisation tasks. Specifically, the course teaches students to build deep neural network architectures and algorithms for solving complex data analysis tasks. In addition to explaining the components, the course also offers a theoretical basis, and addresses objective functions, optimisation algorithms and practical examples for reproducing results. The course emphasises acquiring academic skills such as critical thinking and writing an academic report. To that end, the course comprises two assignments. The first assignment is individual and involves writing a review of an academic article. In the second assignment, students carry out a project assignment on various deep learning architectures and write an academic report with background, methods and results.

Learning objectives

After studying this course, you can:

- distinguish the relative strength of neural network architectures, choose the most suitable architecture for a particular application context, and adapt and prepare data sources for analysis with the chosen architectures;
- adapt, modify and assemble neural network architectures by combining layers of neurons with specific activations, and by adapting their objective function so that it fits the aim of the task in question optimally;
- critically assess research articles on deep learning models, identify positive and negative aspects of the proposed solutions from a technical perspective, by recognising problems with transparency and explainability and by making suggestions for possible improvements and new experiments;
- build advanced neural network architectures (Seq2Seq, Autoencoders, GANs, Object Detection, Graph Embeddings, Large Language Models);
- work in a group on a practical deep learning project by analysing open data, defining the feasibility of a project idea, and by writing an academic report including an evaluation of the feasibility of the solution, as well as a discussion of its explainability and transparency.

IM1112 - Deep Neural Engineering

Type: Standard product **Language:** Dutch **ECTS:** 5 **Price:** € 410.00

The course focuses on neural network methods and models that deal with complex prediction and optimisation tasks. Specifically, the course teaches students to build deep neural network architectures and algorithms for solving complex data analysis tasks. In addition to explaining the components, the course also offers a theoretical basis, and addresses objective functions, optimisation algorithms and practical examples for reproducing results. The course emphasises acquiring academic skills such as critical thinking and writing an academic report. To that end, the course comprises two assignments. The first assignment is individual and involves writing a review of an academic article. In the second assignment, students carry out a project

assignment on various deep learning architectures and write an academic report with background, methods and results.

Learning objectives

After studying this course, you can:

- distinguish the relative strength of neural network architectures, choose the most suitable architecture for a particular application context, and adapt and prepare data sources for analysis with the chosen architectures;
- adapt, modify and assemble neural network architectures by combining layers of neurons with specific activations, and by adapting their objective function so that it fits the aim of the task in question optimally;
- critically assess research articles on deep learning models, identify positive and negative aspects of the proposed solutions from a technical perspective, by recognising problems with transparency and explainability and by making suggestions for possible improvements and new experiments;
- build advanced neural network architectures (Seq2Seq, Autoencoders, GANs, Object Detection, Graph Embeddings, Large Language Models);
- work in a group on a practical deep learning project by analysing open data, defining the feasibility of a project idea, and by writing an academic report including an evaluation of the feasibility of the solution, as well as a discussion of its explainability and transparency.

IM1202 - Model-Based Artificial Intelligence

Type: Standard product **Language:** Dutch **ECTS:** 5 **Price:** € 410.00

Model-based techniques in AI use logic, rules, dynamic models, architectures and more to compute knowledge-based solutions. In contrast to purely data-driven AI, model-based solutions start from formalised knowledge that can be used to find solutions, to speed up the search for solutions, and afterwards to reason with the help of knowledge about the solutions found (as in diagnoses and in explaining predictions made by machine algorithms). The course combines the theoretical study of logic programming with attention to modelling knowledge using logic programs and the practical use of these models for reasoning about dynamic domains, planning and more. Various (programming) assignments in the course contribute to being able to apply the concepts and theories you have learned to concrete problems.

Learning objectives

After studying this course, you can:

- recognise and characterise the role of model-based techniques in AI, and contrast these techniques with data-oriented approaches based on machine learning in terms of knowledge representation, transparency and applicability;
- reason about the theoretical concepts behind ASP, and you have the skills to prove mathematical properties of answer sets;
- name and analyse, from a theoretical perspective, elements of efficient algorithms for ASP;
- you have gained practical skills in modelling computational problems, such as combinatorial problems, as an answer set program;
- determine the suitability of ASP for arriving at an AI solution on the basis of the properties of a realistic problem;
- explain in technical detail how ASP solvers are used to solve problems from other areas of AI.

IM1312 - Research Methods for Artificial Intelligence

Type: Standard product **Language:** Dutch **ECTS:** 5 **Price:** € 410.00

This course is about research methodology for AI and about academic skills. In AI, the methodology has two aspects: a design phase (comparable to that of computer science) and an empirical phase (comparable to that of psychology). The idea is that an AI project first makes something, an artefact, according to certain requirements, and that the artefact (a robot, or a computer program) is then studied in a user context. This course covers a wide range of subjects around this research methodology, including the 'engineering cycle', formulating research questions, designing research, searching for literature, data visualisation, statistics and the ethics of doing research. In addition to working through this material with guidance, in this course you also do a mini-project (your choice: in a group or alone) and go through all the steps of a research project, choosing a topic from AI (multi-agent simulations or computer vision) or human cognition (data visualisation or website design). In addition to an open-book exam in which you have to apply the course material to small examples from the literature, you will also be asked to submit a short research proposal and report on your mini-project and to carry out a peer review of another mini-project's proposal from the course.

Learning objectives

After studying this course, you can:

- describe the characteristics and steps of quantitative (and qualitative) research using the correct terminology, and describe how a design phase can complement the research process in AI research;
- distinguish general empirical cycles in AI research and, in addition, distinguish other cycles, such as design and implementation, that occur mainly in AI, data science and computer science, name the respective sub-steps and apply them to your own research project, and substantiate these with a literature study;
- identify the strengths and weaknesses of different methods and choose the best method for a particular research assignment, and evaluate decisions made in the design of experimental and non-experimental methods;
- determine the measurement level and the role of variables in a study;
- name important ethical issues in research, and evaluate ethical issues in research from the perspective of different stakeholders;
- characterise the reliability and validity of a research project;
- interpret research data, analyse it in the light of a research question and experimental set-up, and report on the research design, the experiments, the results and the conclusions drawn from them.

IM1313 - Information Security Management

Type: Standard product **Language:** Dutch **ECTS:** 7.5 **Price:** € 615.00

Both at the global level and at the level of individual organisations, information security is an increasingly important but also an increasingly complex subject. The constant threat of attacks on information systems forces organisations to devote more and more attention and money to planning and managing the security of these systems. But information security is certainly not only a matter for the management and the IT department of an organisation. In the current information age, it is very important that all employees of the organisation are familiar with the most important aspects of information security, and that they are able to apply the principles of information security in their daily practice. The course 'information security management' studies the management aspects of information security, with an emphasis on governance and compliance. To that end, the current literature is covered on themes such as the importance and development of an information security policy and the role of security awareness programmes, the design of adequate management of information security risks, the most important models and frameworks for

information security management, the measuring and managing of information security performance, and planning for emergency scenarios (incident response, disaster recovery and business continuity). The relevant laws and regulations around information security are also discussed. This Master's course not only emphasises studying the relevant literature, but also devotes ample attention to its practical application in concrete case situations.

Learning objectives

After completing this course, you can:

- understand and describe the most important characteristics of information security, the most important categories of threats to information security and the role of the manager in the context of information security;
- reflect on the role of laws, regulations and ethics in the context of information security and discuss the challenges and methods of digital forensic investigation;
- discuss strategic planning for information security, describe its most important components and make the link with general IT strategic planning and information security;
- design and implement an education, training and awareness programme (SETA) for information security and assess an organisation's information security policy and SETA programme;
- identify, assess and prioritise information security risks and propose an action plan aimed at controlling the relevant risks;
- support organisations in designing an information security approach and suitable information security controls on the basis of (1) their intrinsic characteristics and (2) their operational impact on the organisation, classifying [...]

IM1412 - Generative Artificial Intelligence

Type: Standard product **Language:** Dutch **ECTS:** 5 **Price:** € 410.00

This course focuses on one of the most emerging AI paradigms: Generative AI. The course discusses state-of-the-art Generative AI techniques and models that focus on images, text, music and multimodality, at both a theoretical and a practical level. You study, summarise and present the content of a number of articles to fellow students, with the result that you gain a broad overview of this important sub-field of AI. In the second part of the course, a small research project is carried out in a group. In addition to becoming acquainted with generative AI, in this course you also learn to independently study a new sub-field in the rapidly changing AI landscape.

Learning objectives

After completing the course, you can:

- explain the fundamental concepts and principles of generative AI models;
- describe and analyse key aspects and methods used in building and evaluating Generative AI models for producing solutions in the areas of images, text, music and multimodality, taking into account theoretical as well as practical technical and ethical perspectives;
- search for and select relevant academic literature for a specific Generative AI topic, taking into account given guidelines and criteria;
- develop a video presentation for a specific category and topic of Generative AI, based on your own reflection on academic literature and given requirements for the content and form of the video;
- reflect on a presentation of academic literature by others and give them relevant and constructive feedback;
- present techniques and methods used for a specific Generative AI topic and apply them in the context of a group project;

- present the results of a practical project that makes use of knowledge or results from a state-of-the-art academic article.

IM1903 - Digital Governance

Type: Standard product **Language:** Dutch **ECTS:** 7.5 **Price:** € 615.00

Organisations depend to a large extent on information technology (IT) and data for creating value and achieving competitive advantage. As a result, the role of IT and data is often on the agenda of the management team, and there is increasing attention to IT and data governance. These activities focus in particular on setting up procedures and structures that link the deployment of IT to good alignment with the business strategy, guarantee the quality and integrity of data, and create business value from IT and data. Exactly how IT and data governance is set up depends strongly on the specific role that IT and data play within each individual organisation. In this course, you study various aspects of digital governance (with a focus on IT and data) and develop expertise on the basis of which you can assess the quality of an existing digital governance approach. This course takes a research-based and practice-oriented approach in which you build up your competences step by step. You begin by deepening your theoretical basis by designing an academically substantiated governance model for a (fictional) case, in which you show that you understand the core concepts accurately. You then apply this knowledge in a real context by working in teams of four students with an existing organisation. In doing so, you design a data collection plan, collect data in an ethical way and systematically analyse the digital governance approach in order to arrive at a professional assurance report. Finally, you personalise your learning process by drawing up an in-depth individual advisory report for the organisation. In it, you translate the research results into recommendations firmly rooted in the relevant academic literature.

Learning objectives

In this course, you learn to design and justify a theoretically substantiated digital governance model for a fictional case. In addition, you draw up an academically sound data collection plan that is directly linked to relevant aspects of digital governance, after which you carry out a systematic analysis of these aspects within a specific organisation. Throughout this process, you report on the results of the analysis in a transparent and academic way. In doing so, you critically map out the problems and opportunities for improvement within the organisation. Finally, you translate these insights into practice by formulating appropriate and well-substantiated recommendations for management in the form of a professional advisory report.

IM2203 - Software Engineering and AI

Type: Standard product **Language:** English **ECTS:** 7.5 **Price:** € 615.00

LLMs took many disciplines by surprise, since 2022 there has been an explosion of models, almost following Moore’s law for transistors. These models grew from being a handful to several thousands in no time. Many of these models, ChatGPT and Claude Sonnet included, have the capability of producing code, interacting with code and reason about code. The practice of building systems, including systems that include AI components, is therefore facing the challenge of introducing new tools, new methodologies even new architectures in the practice of software engineering, almost on daily basis. Some clarity on the process a software engineer could follow is needed, from a pedagogical perspective, also, in principles to be effective at using these tools in the most appropriate way, to save time, but also to prevent potentially harmful mistakes that would present only at the very deployment of the system in a production environment. This course builds, or “compiles”, therefore AI into the existing practices of software engineering from a general modelling perspective when building a new system without AI, but using AI tools to build it, and from an architectural perspective, when the AI component is included in the system as a source of information for the functioning of the system. The

emphasis in this course is put on practical, industrial aspects concerning the deployment of AI models, including ethical, qualitative and maintenance issues associated with machine learning in production and potentially interacting wit humans.

Learning objectives

O1: Understand the intersection between data science, machine learning and software engineering and how to make these disciplines work together to deploy software solutionsO2: Use AI Tools to Enhance Software Development software engineering workflows, focusing on improving efficiency and minimizing errors.O3: Acquire knowledge about prompt engineering and using large language models in software engineering practices. O4: Understand design patterns when combined with advanced AI models such as LLMs and object detectors and information retrieval engines.O5: Learn about current law and ethical standards that Software products must comply to.

IM2303 - Security and AI

Type: Standard product **Language:** Dutch **ECTS:** 7.5 **Price:** € 615.00

In this course, you are introduced to two important aspects of cybersecurity and AI. You learn how various AI techniques are applied to tackle a wide variety of cybersecurity problems and challenges. In addition, you learn how AI systems themselves are vulnerable to security and privacy threats and the associated attacks.

IM2403 - Software design and architecture

Type: Standard product **Language:** Dutch **ECTS:** 7.5 **Price:** € 615.00

This course consists of three coherent parts. The first part is about software architecture. A software architect creates a system blueprint on the basis of the sometimes conflicting requirements of stakeholders, with special attention to non-functional requirements such as performance, security and flexibility. In this course, you learn to make architectural choices using architectural patterns, web services and software product lines. You gain insight into how to draw up an architecture description and how to analyse and assess existing software architectures. This part focuses on the adaptability and extensibility of software and applications at a higher level. In the next part, we look at adaptability and extensibility at a lower level, namely of packages and classes in an OO system. Flexibility is an important quality attribute of software and software architecture. The second part of the course focuses on this aspect. Software is never ‘finished’; flexibility with a view to future changes is therefore essential. In this course, you learn and practise design principles for future-proof and flexible software. The third part of the course focuses on academic writing skills. You learn to formulate good research questions, to apply the research process and different research strategies, and to write a clear abstract, introduction and research design for an academic article.

Learning objectives

After completing this course, you can:

- distinguish and describe the stakeholders for a system to be designed;
- describe the stakeholders’ concerns from their point of view;
- draw up requirements from the concerns, devising a solution for conflicting concerns;
- use architectural patterns and styles to design a system that meets the requirements;
- argue the advantages and disadvantages of design decisions;
- write an architecture document that can be understood by all stakeholders;
- analyse and evaluate a given architecture;
- indicate which problems can occur when combining components into a larger software system, and offer solutions;

- explain the following concepts: standard architectures, quality models (ISO 25010), architectural patterns, architectural models (IEEE 1471, UML and others), assessing an architecture, REST, software product lines, variability and composition;
- have insight into the meaning of software modifiability and its importance for designing software;
- have knowledge of design principles, design tactics, the onion architecture, MVC and domain-driven design as approaches to support software modifiability;
- be able to apply design patterns as concrete solutions to support design principles and design tactics for modifiability [...]

IM9502 - AI Graduation Assignment Preparation

Type: Standard product **Language:** Dutch **ECTS:** 5 **Price:** € 410.00

In this course, you draw up a research plan in consultation with the graduation committee. The aim is to arrive at well-substantiated research questions and a plan of action to answer these questions, which is carried out in the graduation project itself (IM9506). An important part of this course is a literature study in which the research question is placed within the existing literature and in which insights from that literature are used to answer the question as well as possible. The end product of this course is a research proposal that is assessed on, among other things, the academic relevance and feasibility of the research. During your graduation, you have the freedom to choose different types of research. You could build an AI system to solve a specific problem, and validate this system empirically in terms of performance on a task or the consequences of the system in a particular domain. You could also work on a new AI algorithm to solve a particular task, exploring the capabilities of this new algorithm. Finally, it would even be possible to do more theoretical research into techniques or phenomena, using mathematical models or other formal tools.

Learning objectives

After completing this course, you can:

- formulate sensible and responsible research questions on the basis of a chosen assignment and literature study;
- place academic literature in the right context by describing a logical coherence between articles;
- select a suitable research method on the basis of research questions and describe it in sufficient detail;
- divide a study into phases and describe and plan those phases;
- justify choices in a research plan.

IM9506 - Artificial Intelligence Graduation Assignment

Type: Standard product **Language:** Dutch **ECTS:** 15 **Price:** € 1,230.00

In this course, you carry out the project that was proposed (and approved) in the course AI Graduation Assignment Preparation. In this course, all content-related and methodological aspects of the Master's programme come together in a project in which you carry out all parts of an AI study.

Learning objectives

After completing the course, you can:

- develop the experimental environment needed to be able to answer a research question;
- apply subject-specific and methodological knowledge during the research;
- integrate knowledge from different sub-fields of AI;
- abstract a practical problem to an academic level, and translate the results of academic research back into a solution for the problem;
- give a well-founded vision of your own research;

- justify choices made regarding approach and result, with an eye for reliability and validity;
- reflect on ethical aspects of your own research;
- analyse, interpret and evaluate the results found from the assignment carried out;
- evaluate the contribution of the assignment carried out to the development of the field from which the question arose;
- formulate responsible conclusions regarding the results found;
- formulate sensible recommendations for further research;
- report in writing and orally on the results of the research.

IM9605 - CS Graduation assignment preparation

Type: Standard product **Language:** Dutch **ECTS:** 12.5 **Price:** € 1,025.00

During this module, you show that, through literature study and possibly preliminary research, you can, for a chosen assignment:

- formulate it as a research assignment;
- place it in a context;
- provide it with a suitable intended approach and solution method;
- divide it into phases and describe and plan these phases;
- account for the choices made.

You start the preparation of the graduation track with an orientation. You read the graduation guide: it is made available digitally via the digital learning environment. You can also consult the digital learning environment for information on subjects, graduation supervisors, the graduation process and examples of graduation assignments. You can also attend graduation meetings and graduation presentations. The next step is to make contact with a graduation supervisor. After an orientation conversation, you make your choice known by completing the online registration form Graduation track CS. You are assigned a graduation committee and receive instructions to register for IM9605. During the preparation of your graduation assignment, it may become clear that studying additional material is necessary. In consultation with your graduation committee, it is determined which in-depth study in a particular computer science direction, or which specific domain knowledge of the domain within which your research takes place, is necessary. You use this in-depth study when writing your research plan. After your research plan has been approved and assessed by your graduation committee, you can continue the graduation track with IM990C Computer Science Graduation Assignment.

IM9606 - CS Graduation assignment preparation

Type: Standard product **Language:** Dutch **ECTS:** 15 **Price:** € 1,230.00

During this module, you show that, through literature study and possibly preliminary research, you can, for a chosen assignment:

- formulate it as a research assignment;
- place it in a context;
- provide it with a suitable intended approach and solution method;
- divide it into phases and describe and plan these phases;
- account for the choices made.

You start the preparation of the graduation track with an orientation. You read the graduation guide: it is made available digitally via the digital learning environment. You can also consult the digital learning environment for information on subjects, graduation supervisors, the graduation process and examples of graduation assignments. You can also attend graduation meetings and graduation presentations. The next

step is to make contact with a graduation supervisor. After an orientation conversation, you make your choice known by completing the online registration form Graduation track CS. You are assigned a graduation committee and receive instructions to register for IM9606. During the preparation of your graduation assignment, it may become clear that studying additional material is necessary. In consultation with your graduation committee, it is determined which in-depth study in a particular computer science direction, or which specific domain knowledge of the domain within which your research takes place, is necessary. You use this in-depth study when writing your research plan. After your research plan has been approved and assessed by your graduation committee, you can continue the graduation track with IM990C Computer Science Graduation Assignment.

IM9703 - SE Graduation assignment preparation

Type: Standard product **Language:** Dutch **ECTS:** 7.5 **Price:** € 615.00

This course starts every quartile. In addition to individual supervision by the graduation supervisor, you take part in the supervision meetings and the graduation meetings.

- You work largely independently, but stay in contact with your graduation supervisor via e-mail and (online) consultation. You attend the three supervision meetings during the quartile. In these, the planning and progress are discussed and you exchange experiences with fellow graduating students.
- You attend the graduation meetings, which are held five times a year (at the start of each quartile, and in July). In these meetings, you contribute to discussions, give feedback to your fellow students, report on your progress and give two presentations about your work.
- You receive feedback on interim results (usually after they have been put in writing or presented at a graduation meeting).

IM9906 - Software Engineering graduation assignment

Dutch name: Afstudeeropdracht Software Engineering **Type:** Standard product **Language:** Dutch **ECTS:** 15 **Price:** € 1,230.00

Learning objectives

In this phase of the graduation track, you carry out the agreed activities as approved by your graduation committee. After completing this course, you can:

- develop software to support answering your research question;
- apply subject-specific and methodological knowledge;
- integrate knowledge from different (sub)fields;
- abstract the practical situation to a generic solution and translate the results back into academic terms;
- develop solutions to formulated problems;
- formulate a well-founded vision of your own;
- account for choices made regarding approach and result, with an eye for reliability and validity;
- design in a responsible way with reliable technologies;
- analyse, interpret and evaluate the results found and the assignment carried out;
- evaluate the contribution of the assignment carried out to the development of the field from which the question arose;
- formulate conclusions regarding the results found;
- formulate recommendations for further research;
- report in writing and orally on the results of the assignment;
- work under the conditions of the Netherlands Code of Conduct for Research Integrity.

IM990C - Computer Science graduation assignment

Dutch name: Afstudeeropdracht Computer Science **Type:** Standard product **Language:** Dutch **ECTS:** 30
Price: € 2,460.00

In this phase of the graduation track, you carry out the agreed activities as approved by your graduation committee. In working out your research plan, you show that you have acquired the following knowledge and skills and can:

- apply subject-specific and methodological knowledge;
- integrate knowledge from different (sub)fields;
- abstract the practical situation to a generic solution and translate the results back into academic terms;
- develop solutions to formulated problems;
- formulate a well-founded vision of your own;
- account for choices made regarding approach and result, with an eye for reliability and validity;
- design in a responsible way with reliable technologies;
- analyse, interpret and evaluate the results found and the assignment carried out;
- evaluate the contribution of the assignment carried out to the development of the field from which the question arose;
- formulate conclusions regarding the results found;
- formulate recommendations for further research;
- report in writing and orally on the results of the assignment;
- work under the conditions of the Netherlands Code of Conduct for Research Integrity.

MAAI-2026-2027 - Master's programme in Artificial Intelligence

Dutch name: Master Artificial Intelligence **Type:** Training **Language:** Dutch **ECTS:** 60 **Price:** -

The Master’s programme in AI covers modern, academic methods for model-oriented and data-oriented AI, with an emphasis on social and ethical aspects. You receive a broad introduction to the academic field of AI and to the academic skills and methods for AI research. In addition, you can deepen your academic knowledge in sub-fields through a capita selecta course and the graduation track. The Master’s programme is divided into four themes: the AI field, data-oriented techniques, model-oriented techniques and new AI developments.

The AI field

The first part of the programme covers the AI field in the broadest sense of the word and forms the basis of this Master’s programme. You learn how AI systems are designed and studied systematically and scientifically. You also delve into the ethical consequences of algorithms, and how you can technically solve certain issues around privacy, freedom and security.

Data-oriented techniques

Using data-oriented techniques such as Machine Learning and Deep Learning, you learn, among other things, to design systems that can make predictions on the basis of data. There is ample attention to the technical skills for applying and creating Machine Learning models yourself. You also acquire the knowledge and practical skills to learn from data with deep neural networks.

Model-oriented techniques

With model-based techniques, you learn to develop knowledge-based AI systems. You do this by recording knowledge in languages based on logic and probability theory. You learn to recognise the role of model-based techniques in AI and to compare them with data-oriented approaches. The underlying theoretical concepts are covered, and you acquire skills for modelling knowledge and the practical use of these models.

New AI developments

In the course Generative Artificial Intelligence, you study a new, current sub-field within the rapidly changing AI landscape. Guided by an AI researcher, you complete the Master's in AI with a thesis and a final presentation.

Do you want to know more about the content? Request the study guide.

MACS-2026-2027 - Master's programme in Computer Science (MSc)

Dutch name: Master Computer Science (MSc) **Type:** Training **Language:** Dutch **ECTS:** 120 **Price:** -

This Master's programme is a broad programme at the intersection of information technology, software engineering, artificial intelligence and business administration. The programme focuses on sub-fields of computer science that are important for professional practice. Depth, specialisation and academic skills are central to the programme. You are given a large role and responsibility yourself in finding, comparing, assessing and applying academic insights to current issues from practice or from research at the Open Universiteit.

Structure

In the Master's in Computer Science, theory and application in practice go hand in hand. Many of the subjects covered are ones you recognise from your own professional practice. The programme offers a unique combination of the fields of information technology, software engineering, artificial intelligence and business administration. The programme consists of a compulsory part in which you acquire elementary knowledge of software engineering, artificial intelligence, information systems and business processes. You can then choose an elective part for further specialisation in an area that fits your own educational needs, work experience and interests. You can choose from the elective areas Artificial Intelligence, Software Engineering, Security or Information Science. In business, there is great demand for trained computer scientists. The broad character and focus of the programme ensure a good match between the programme and the labour market. Software systems are becoming ever larger and more complex. The role of data science and artificial intelligence is increasing strongly. Companies therefore need people who have a good overview of the field and who are able to independently keep up with new developments and assess their relevance for the company. At the same time, companies and government bodies are struggling with complex (automation) projects and the challenges that technological innovations bring with them. All this calls for up-to-date academic education. In the programme, you develop the academic skills needed to carry out research independently. You practise these skills through writing assignments, peer reviews or giving a presentation. You complete your studies with a graduation assignment, in which research and practice come together.

Do you want to know more about the content? Request the study guide.

MAKSE - Master's programme Software Engineering, classroom-supervised

Dutch name: Master Software Engineering klassikaal begeleid **Type:** Standard product **Language:** Dutch **ECTS:** 75 **Price:** € 0.00

The university Master's programme in Software Engineering is offered in the classroom-based form of tuition in cooperation with Fontys Hogescholen. Software engineering is a discipline that is constantly on the move and, owing to turbulent developments in all kinds of areas, constantly faces new challenges. Software systems are becoming ever larger and more complex. More and more functions of the devices we use every day, from cars to microwaves, are realised partly in hardware and partly in software. Programming languages and design disciplines that were adequate fifteen years ago are now hopelessly outdated. Companies engaged in software development therefore need people who not only have a good overview of the field, but are also

able to independently keep up with new developments and assess their relevance for the company. This calls for academic education, which is offered in the academic Master's programme in Software Engineering. At the heart of the curriculum are the development and maintenance of high-quality software, with attention to both the process side and the technical side. Three sub-aspects are distinguished: Software architecture: focuses on the high-level design of well-structured and easily maintainable software. Software quality management: focuses on quality in the phases of the software lifecycle and managing it, both at the start when determining the requirements and later when the software evolves. Software quality assurance: focuses on (formal) techniques for quality monitoring and quality determination, with regard to both functional and non-functional properties. The programme consists of the Premaster Formal Techniques (bridging programme, 15 EC credits) and the courses of the Master's programme in Software Engineering (60 EC credits). At the heart of the curriculum are the development and maintenance of high-quality software, with attention to both the process side and the technical side. Three sub-aspects are distinguished: software architecture, software quality management and software quality assurance. The structure of the programme is as follows:

Programme content

Premaster Formal Techniques

Before they can be admitted to the Master's programme, students with a closely related higher professional education (HBO) Bachelor's degree, such as computer science, technical computer science or business informatics, have to complete a bridging programme of 15 EC in which data structures and algorithms, formal languages, logic, programming languages and security are covered. If you have a prior education other than a related HBO Bachelor's degree, a longer premaster or direct admission to the Master's programme may apply [...]

MASE-2026-2027 - Master's programme in Software Engineering (MSc)

Dutch name: Master Software Engineering (MSc) **Type:** Training **Language:** Dutch **ECTS:** 60 **Price:** -

Software Engineering is a discipline that is constantly on the move and, owing to rapid developments in hardware and software, constantly faces new challenges. Software systems are becoming ever larger and more complex. There is therefore a need for people who not only have a good overview of the field, but who are also able to independently keep up with new developments and assess their relevance for the organisation concerned. At the heart of this programme are the development and maintenance of high-quality software, with attention to both the process side and the technical side. We distinguish three sub-aspects: software architecture, software quality management and software quality assurance. In the Master's programme, the emphasis is on both research and practice, because in practice too there is great demand in business for academically trained software engineers. Many IT projects already fail in the design phase. In the programme, you learn to improve and extend existing software through better software maintenance and renovation at an academic level (course Software Quality Management) and better verification of security (course Software Security) and better testing (course System Verification and Testing). You also learn to develop new software better through a better software definition: development and verification at an academic level prevents problems (course Software Design and Architecture). In addition, you learn to carry out management better by making more efficient use of IT-specific management techniques (course Software Quality Management). The programme is not only an asset for practice, but also in an academic sense. Academic skills such as academic writing and research skills are covered in an integrated way in the courses. The graduation assignment is par excellence the point where both practical and research skills matter. A respectable number of our students have, at graduation, a research result such that they publish an academic article about it together with the lecturer.

Do you want to know more about the content? Request the study guide.

MODL01 - Developing and delivering online and blended learning

Dutch name: Ontwikkelen en verzorgen van online en blended leren **Type:** Simple product **Language:** Dutch **Price:** € 0.00

By enrolling in this series of online knowledge sessions, you get free access to seven knowledge sessions, which you can complete at your own pace and at any time: Online visibility, Social media advertising, Online safety, Data analysis, Data analysis process, Text analysis, Data visualisation.

MODL02 - Developing and delivering online and blended learning

Dutch name: Ontwikkelen en verzorgen van online en blended leren **Type:** Simple product **Language:** Dutch **Price:** € 0.00

By enrolling in this series of online knowledge sessions, you get free access to eleven knowledge sessions, which you can complete at your own pace and at any time: Introduction to digitalisation in logistics, Digitalisation in the purchasing process, Digitalisation in the production process, Digitalisation in the warehousing process, Digitalisation in the transport process, Digitalisation in return logistics, What do I use as a lever to digitalise?, Making a business case, Make or Buy decisions, The implementation, Best practice example.

OBAINF-2026-2027 - Open Bachelor's programme in Computer Science

Dutch name: Open bachelor informatica **Type:** Training **Language:** Dutch **ECTS:** 180 **Price:** -

This programme is a combination of courses from the academic field of Computer Science with broadening courses from one or two other academic fields. There are various options in broadening packages. The combination of two or even three fields of study gives the programme a multidisciplinary character.

SMAAI-2026-2027 - Bridging programme Artificial Intelligence

Dutch name: Schakelprogramma Artificial Intelligence **Type:** Training **Language:** Dutch **ECTS:** 5 **Price:** -

The premaster Artificial Intelligence (AI) is a combination of a maximum of five of the following courses:

- IB0402 - Logic, sets and relations (5 EC)
- IB0602 - Linear algebra and stochastics (5 EC)
- IB3502 - Premaster AI: programming with Python (5 EC)
- IB3702 - Premaster AI: mathematics for Machine learning (5 EC)

The size is therefore a maximum of 20 credits.

SMACS-2026-2027 - Bridging programme Computer Science

Dutch name: Schakelprogramma Computer Science **Type:** Training **Language:** Dutch **ECTS:** 0 **Price:** -

The premaster is a combination of a maximum of eight courses. At admission, a list of subjects is used that you must demonstrably master on the basis of your prior education. For a subject that you do not master, a premaster course is prescribed. The subjects are intended to ensure that your prior knowledge is up to standard. The premaster courses each have a size of 5 EC:

- IB1102 Object-oriented programming
- IB1002 Object-oriented analysis and design
- IB0402 Logic, sets and relations

- IB0602 Linear algebra and stochastics
- IB0802 Formal languages and automata
- IB1812 Security and IT
- IB2702 Concepts of programming languages
- IB3702 Mathematics for Machine Learning

Do you want to know more? Request the study guide.

SMASE-2026-2027 - Bridging programme Software Engineering

Dutch name: Schakelprogramma Software Engineering **Type:** Training **Language:** Dutch **ECTS:** 0 **Price:** -

The premaster is a combination of six courses of 5 EC each. These six courses are: IB1102 Object-oriented programming, IB1002 Object-oriented analysis and design, IB0402 Logic, sets and relations, IB0802 Formal languages and automata, IB1812 Security and IT and IB2702 Concepts of programming languages.

Do you want to know more? Request the study guide.