General Information

Audience: Bachelor ICT 3rd year with experience in programming.
15 European Credits (10 weeks)

When

September – November
Class days: 3 to 4 days (1 day off)

Teachers
Teachers of the Computer Science Haarlem study program and guest lectures and workshops by specialists from the field of cloud computing.

Details of assessments

- Project assessment consisting of a specification review, code review and two presentations.
- Individual code assessment
- Individual server side programming assignment
- Written exam on cloud computing theory

All assessments must be completed with a sufficient grade

Contact

Erwin de Vries, Haarlem
erwin.devries@inholland.nl

inholland
hogeschool

Domain TOI – Cluster ICT
Minor
Cloud computing

Cloud computing

Over the past decade server side software deployment models have rapidly changed, and the availability of development services has greatly increased. Thorough knowledge of IaaS and PaaS solutions is vital to software developers specializing in backend software development. This minor addresses these topics in a hands on manner, and teaches you how to develop *serverless* solutions. It consists of various classes and workshops with topics ranging from obtaining hands on experience with Microsoft Azure, to cloud database models, REST API design and testing, Continuous Integration / Continuous Delivery (CI/CD) and DevOps.

A general overview of the scope of the concepts of cloud computing is provided by the Cloud Computing course. The API Design course teaches you how to properly design and document an API based on the RESTful paradigm using OpenAPI (Swagger). The API Testing course provides you insights in how to perform automated tests on an API, and how to integrate this into a CI/CD pipeline. A course in Cloud Databases allows you to differentiate between various options at hand for storing information in a scalable manner. Hands on experience is provided by several Azure development workshops.

The core of the minor consists of a group project for an external client (not for profit), which provides you the opportunity to seeing your backend being applied in practice.

Goals

The student is able to:

- identify cloud deployment models and apply these models in a development process
- design an API using the RESTful paradigm
- model and document an API using OpenAPI (Swagger)
- differentiate between various cloud database models, and apply them to a software solution
- design and develop highly scalable serverless cloud solutions
- cooperate with fellow students in software development activities
- effectively communicate with external clients (not for profit)

Strategies and teaching activities

- Workshops by experts
- Do research with your project group
- Lectures on theory combined with practical exercises

Competences

- Management - Applying principles for managing and safeguarding a software development process
- Management - Setting up and executing management of a public of private cloud based infrastructure
- Management - Applying configuration- change, and release management
- Analysing - Exploring integration and migration difficulties
- Analysing - Specifying a distributed computer system consisting of timing, resource usage and performance
- Designing - Setting up a technical design for an infrastructure including the appropriate security, based on functional and non-functional requirements

- Designing - Designing a software system while taking into account existing components and libraries, while making use of design principles and/or quality criteria
- Designing - Designing a cloud based infrastructure while taking note of all requirements
- Designing - Setting up a test strategy for system tests
- Designing - Designing a distributed computer system, including setting up actuators, sensors, timing, resource usage and performance
- Realising - Setting up an infrastructure that meets demands in terms of performance, usability, security and compliance
- Realising - Realising a public or private cloud based infrastructure and services while taking note of all requirements
- Realising - Application of automated tests
- Realising - Designing and applying an acceptation procedure, for example in a virtualized environment, including aspects related to timing, resource usage and performance