

# Teaching Vision

---

## Teaching Philosophy.....

My teaching philosophy is rooted in the conviction that effective education hinges on a deep comprehension of theoretical concepts coupled with their practical implementation. I adopt a student-focused approach, prioritizing interactive learning and problem-solving in real-world contexts. The fundamental role of an educator, as I perceive it, is to endow students with the skills necessary for addressing real-world challenges. This involves not only imparting conceptual knowledge—'what to think'—but also cultivating the ability to think critically and collaborate effectively in implementing scalable solutions. Particularly for university graduates entering industry or academia, competencies in teamwork, practical systems design and implementation, and scientific communication are indispensable. My courses are aligned with these objectives. Drawing from my experience in both industry and academia, I believe I am well positioned to develop and deliver curricula that are both relevant and enriching.

## Course design and curriculum development.....

1. Concepts of Programming Languages: In my proposed course, drawing upon my rich teaching background in programming languages at TU Darmstadt, I aim to provide a comprehensive exploration of the fundamental concepts of programming languages. This course will span a spectrum of topics, from the rudiments of compilation and variables to the complexities of conditionals and subprograms. It will be meticulously crafted in a language-neutral manner, employing illustrative examples from a variety of programming languages to effectively elucidate each concept. A unique feature of this course will be an additional, stimulating component where students will progressively develop a programming language, integrating features introduced in each lecture. This hands-on aspect will utilize the modeling language workbench known as MPS from JetBrains. To enrich the learning journey, I plan to incorporate resources from my own video playlist, offering a diverse, multimedia educational experience. This course is ideally suited for **BSc** students. For those at the **MSc** level, I intend to include further study into the fundamentals of static analysis, covering topics such as Lattices, Monotone Framework, Control Flow Graph, and Data-Flow Analysis. The course outcomes are designed to be both **conceptual** and **practical**, with a special focus on **team-building** skills, a critical competency in the professional sphere. The course materials, already available in the form of slides and video recordings, will be tailored to meet the specific needs of Utrecht University. The recommended literature for this course is *Robert Sebesta's Concepts of Programming Languages*.
2. Scientific Writing for computer science: Leveraging the success of its previous iterations, this course aims to mentor students in crafting high-caliber vision papers, a number of which have achieved publication in renowned conferences and workshops. The course will commence with an in-depth examination of cutting-edge literature on selected topics, such as 'Artificial Intelligence for Software Assistance'. Following this, we will engage in critical discussions and analyses of novel research ideas building upon this foundational knowledge. Participants will then articulate these ideas in succinct vision papers. Primarily targeting **MSc** candidates, this course is also open to ambitious **BSc** students. The course

is designed to cultivate a host of skills, including **literature consumption**, **scientific writing**, **idea formulation**, and **teamwork**. Predominantly interactive in nature, the course will feature workshops where students will brainstorm ideas, present summaries, and collaborate in teams to compose their vision papers, all under expert guidance. Additionally, a series of lectures will be provided, focusing on the essentials of writing in the field of computer science, based on *Writing for Computer Science* by Justin Zobel.

3. **DSL Engineering:** This course will be designed to provide an in-depth immersion into the theory and practice of Domain-Specific Language (DSL) design and implementation. It comprehensively covers the fundamental aspects of DSLs, highlighting their advantages and potential challenges. Key topics such as modular languages, static semantics, and language composition are explored in depth. Participants will engage in practical, hands-on experiences with various DSL tools including Xtext, MPS, and Spoofox, gaining insights into their application at different stages of DSL development. The course emphasizes real-world applications in software engineering, requirements engineering, and software architecture, blending lectures with project work to equip students with the skills to design, implement, and apply DSLs effectively across various domains. This course is particularly suited for **MSc** students who have completed the 'Concepts of Programming Languages' course and have a keen interest in programming languages and software development methodologies. It aims to develop skills such as **systems thinking** and **model-driven DSL design and implementation**. The primary reference material will be *DSL Engineering* by Markus Voelter.
4. **Others:** With my experience as a co-lecturer in 'Introduction to Software Engineering' at TU Darmstadt, a mandatory course for BSc students, I am well-positioned to contribute significantly to similar courses at Utrecht University. My enthusiasm extends to creating courses that align with the latest advancements in AI. One exciting proposition is a practical course where students employ Large Language Models (LLMs) for software and DSL development. This course could involve innovative approaches like comparing and evaluating strategies for teaching LLMs the grammars of unfamiliar DSLs, examining techniques such as fine-tuning, pre-prompt injection, and post-processing. Overall, I am keen to engage in lecturing opportunities within my area of expertise that align with Utrecht University's vision and the evolving landscape of Software Engineering.

## Conclusion.....

At Utrecht University, I aspire to be more than an educator; I aim to be a mentor and a collaborator, significantly contributing to the department's growth and the students' success. I firmly believe that an educator should also be a perpetual learner. The opportunity for personal and professional growth through the Individual Development Plan, as highlighted in the position, reinforces my belief that this role can foster a mutually enriching relationship. I am enthusiastic about bringing my expertise and learning mindset to Utrecht University, confident that together we can achieve remarkable advancements in both education and research.