Introduction

My Name is Hans de Nivelle, I will be teaching 'Programming Paradigms' to you. I studied in Delft, the Netherlands.

I am also known, in particular to the **registrar** as **Jean de Nivelle**.

From Wikipedia: Johannes is a Medieval Latin form of the personal name that usually appears as "John" in English language contexts. It is a variant of the Greek name $\omega\alpha\nu\nu\eta\varsigma$ and Classical Latin (Ioannes), itself derived from the Hebrew name Yehochanan, meaning "Yahweh is gracious". The name became popular in Northern Europe, especially in Germany. Common German variants for Johannes are Johann, Hans (diminutized to Hänschen or Hänsel, known from "Hansel and Gretel", a fairy tale by the Grimm brothers), Hannes, Jens (from Danish) and Jan (from Dutch). In the Netherlands, Johannes was without interruption the

most common masculine birth name until 1989.[2] Jan is a variant of John known in Catalan, Czech, Slovenian, Dutch, Scandinavian, Cornish, German, Afrikaans and Northern Germanic. Polish has its own variant Janusz as has Slovenian "Janez". Common English variants for Johannes are John or Johnny. There are also variants of the name in other languages:

- I got my PhD from Delft University of Technology in the Netherlands in 1995. Title: Ordering Refinements of Resolution.
- 1999-2007, I worked in Saarbrücken, Germany, at Max-Planck Institute for Computer Science.
- 2007-2017, I worked in Wrocław, Poland, at University of Wrocław.
- October 2017, I started here at NU.

Contents of the Course

The main programming paradigms are imperative programming, functional programming, and logic programming.

I will treat all three during the course, but we will spend little time to functional programming. We will in depth cover two imperative languages (C^{++} and Python), and one logic programming language Prolog. Most common programming languages are imperative. They are closest to electronic hardware, and the most efficient.

The other two programming paradigms have mathematical foundation. They are more suitable for some types of problems, often one can write a program much quicker, but they tend to be less efficient.

Components of Programming Languages

I will also treat the main components of programming languages, these are

- Difference between compiled and interpreted languages.
- Type systems, how to check type correctness. Static versus dynamic typing.
- Value semantics versus reference semantics. Connection to Object-Oriented Programming. Consequences for memory management.

Goals

'Computer Science' is also Science. You will learn something about the S-part of CS.

Know how to choose the proper language for a given task. Know what you can choose from.

Know what are the building blocks of programming languages.

Learn good coding practice. Readability is nearly always more important than efficiency.

Topics

The course will be roughly divided as follows:

- C^{++} : 50%,
- Prolog: 25%,
- Python: 25%.

There will be programming exercies, two midterm exams, one final exam. Wait for the syllabus.

Be prepared to learn Linux (ubuntu).