# Introduction, administrivia

## Data Structures and Algorithms for Computational Linguistics III
## ISCL-BA-07

Çağrı Çöltekin
/tʃaːɾˈɯ tʃœltecˈɪn/
ccoltekin@sfs.uni-tuebingen.de

University of Tübingen
Seminar für Sprachwissenschaft

Winter Semester 2026–2025

version: 87deae7 @2026-10-19

# What is this course about?

- An intermediate-level course on programming
- Algorithms: (good) solutions to programming problems
- Data structures: (efficient) ways to organize/store information

Prerequisites:

- Data Structures and Algorithms for CL I
- Data Structures and Algorithms for CL II

Module: ISCL-BA-07, Advanced Programming

# What is in this course?

## A bird's eye view

Introductory lectures on

- Some fundamental data structures: queues, stacks, trees, graphs, ...
- Some fundamental algorithms: searching, sorting, pattern matching, graph algorithms
- Analysis of algorithms
- Finite state automata
- Parsing

# Why study algorithms?

(in the age of LLMs)

- It is one of the fundamental topics in computer science: an algorithm is the way you instruct a computer to do things
- Knowing a clever, efficient solution to one problem helps designing good solutions for other, related problems
- Learning basic algorithmic techniques makes you a better programmer, better problem solver
- Designing good algorithms is an intellectual challenge
- The most popular interview questions for programming jobs are about algorithms

# Course overview

- Lectures
  - Monday 14:15-15:45 (VG 0.02, Wilhelmstr. 19)
  - Wednesday 14:15-15:45 (VG 0.02, Wilhelmstr. 19)
- Lab: Friday 14:15-17:45 (VG 0.02, Wilhelmstr. 19)
- Tutors:
  - How Chuen Yu (Kyle)
  - Sung-Jin Miriam Han
- Public course website: `https://iscl-dsacl3.github.io/`
- Moodle: `https://moodle.zdv.uni-tuebingen.de/course/view.php?id=1980`
  - Please fill in the 'beginning of semester survey' before the next course session
- GitHub: `https://github.com/iscl-dsacl3/dsacl3-2026`

# Literature

- *Data Structures and Algorithms in Python*. Goodrich, Tamassia, and Goldwasser (2013)
  - Available through university library (online version): `https://ebookcentral.proquest.com/lib/unitueb/detail.action?docID=4946360`
  - Website of the book contains source code, hints, examples: `http://bcs.wiley.com/he-bcs/Books?action=index&bcsId=8029&itemId=1118290275`
- *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Jurafsky and Martin (2009)
  - Draft chapters of 3rd edition is available at `https://web.stanford.edu/~jurafsky/slp3/`
- Course notes will be provided for some topics

# Coursework and evaluation

- Reading material for most lectures
- Weekly assignments: ungraded, but **required**: For successful completion of the practical part of the class, you have to complete all the assignments.
- Final (written) exam (80 %)
- Final project (20 %)
- Attendance is not required, but you are unlikely to pass without regular attendance

# Assignments & Projects

- Assignments in Python
- Only online submissions through GitHub
- The assignments are to be done individually
- Projects are group projects (knowing your classmates, and learning from them, is an important part of the university experience/education)
- Projects will run throughout the semester with strict deadlines
- More information on projects next week Friday during the lab session.

You are recommended **not** to use LLMs for solving the assignments. You will learn more/better if you struggle through them.

# Topics at a glance

- A recap of what you should already know: arrays, lists, maps, queues, stacks, iteration, recursion, binary search, ...
- Algorithmic analysis
- Common algorithmic patterns: brute force, greedy, divide and conquer, dynamic programming, ...
- Sorting
- Trees
- Priority queues, heaps
- Hashing
- Graphs, graph algorithms
- Pattern matching
- Tries
- Finite state automata and regular expressions
- Finite state transducers
- Parsing

# Final remarks

- This course is not about learning programming, or a programming language, it is about solving computational problems
- This is a 12CP course: it means you are expected to spend 360 hours (over 200 hours outside class) studying it
- Please do not be shy, ask your questions during the lectures
- Please take the assignments seriously, learning programming requires practice
- Next:
  - a recap of basic data structures and algorithms
  - Assignment 0

# Final remarks

- This course is not about learning programming, or a programming language, it is about solving computational problems
- This is a 12CP course: it means you are expected to spend 360 hours (over 200 hours outside class) studying it
- Please do not be shy, ask your questions during the lectures
- Please take the assignments seriously, learning programming requires practice
- Next:
  - a recap of basic data structures and algorithms
  - Assignment 0

- Time for your questions

# Acknowledgments, credits, references


- Goodrich, Michael T., Roberto Tamassia, and Michael H. Goldwasser (2013). *Data Structures and Algorithms in Python*. John Wiley & Sons, Incorporated. ISBN: 9781118476734.
- Jurafsky, Daniel and James H. Martin (2009). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. second edition. Pearson Prentice Hall. ISBN: 978-0-13-504196-3.