

Planning and Optimization

A1. Organizational Matters

Malte Helmert and Gabriele Röger

Universität Basel

September 16, 2026

Planning and Optimization

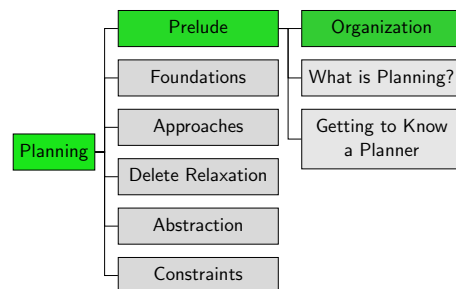
September 16, 2026 — A1. Organizational Matters

A1.1 People & Coordinates

A1.2 Target Audience & Rules

A1.3 Course Content

Content of the Course



A1.1 People & Coordinates

People: Lecturers



Malte Helmert



Gabriele Röger

Lecturers

Malte Helmert

- ▶ **email:** malte.helmert@unibas.ch
- ▶ **office:** room 06.004, Spiegelgasse 1

Gabriele Röger

- ▶ **email:** gabriele.roeger@unibas.ch
- ▶ **office:** room 04.005, Spiegelgasse 1

People: Assistant



David Speck

Assistant

David Speck

- ▶ **email:** davidjakob.speck@unibas.ch
- ▶ **office:** room 04.003, Spiegelgasse 5

People: Tutors



Gustavo Delazeri



Jil Panter



Florian Pommerening

Tutors

Gustavo Delazeri and Jil Panter

- ▶ **email:** {gustavo.delazeri,j.panter}@unibas.ch
- ▶ **office:** room 04.001, Spiegelgasse 5

Florian Pommerening

- ▶ **email:** florian.pommerening@unibas.ch
- ▶ **office:** room 04.005, Spiegelgasse 1

Time & Place

Lectures

- ▶ **time:** Mon 14:15–16:00, Wed 14:15–16:00
- ▶ **place:** room 00.003, Spiegelgasse 1

Exercise Sessions

- ▶ **time:** Wed 16:15–18:00
- ▶ **place:** room 00.003, Spiegelgasse 1

first exercise session: today

Communication Channels

- ▶ lecture sessions (Mon, Wed)
- ▶ exercise sessions (Wed)
- ▶ course homepage
- ▶ ADAM workspace
- ▶ Discord server (invitation link on ADAM workspace)
- ▶ email

registration:

- ▶ <https://services.unibas.ch/>
- ▶ Please register today to receive all course-related emails!

Planning and Optimization Course on the Web

Course Homepage

<https://dmi.unibas.ch/en/studium/computer-science-informatik/lehrrangebot-hs26/lecture-planning-and-optimization/>

- ▶ course information
- ▶ slides
- ▶ link to ADAM workspace
- ▶ bonus materials (not relevant for the exam)

A1.2 Target Audience & Rules

Target Audience (Computer Science “Old”)

computer science students (regulations of spring 2026 and earlier):

- ▶ M.Sc. Computer Science
 - ▶ Major in Machine Intelligence:
 - module [Concepts of Machine Intelligence](#)
 - module [Methods of Machine Intelligence](#)
 - ▶ Major in Distributed Systems:
 - module [Applications of Distributed Systems](#)
- ▶ M.A. Computer Science (“Master-Studienfach”)
 - module [Concepts of Machine Intelligence](#)

Target Audience (Computer Science “New”)

computer science students (regulations of fall 2026 and later):

- ▶ M.Sc. Computer Science and M.A. Computer Science
 - ▶ specialization module [Machine Intelligence](#)
 - ▶ you have a lot of freedom in selecting your courses, make use of it
 - ▶ recommended for students that enjoy mathematical logic and formally rigorous approaches to computer science

Target Audience (Others)

everyone else:

- ▶ M.Sc. Data Science: module [Electives in Data Science](#)
- ▶ other students welcome

Prerequisites

prerequisites:

- ▶ general computer science background: good knowledge of
 - ▶ algorithms and data structures
 - ▶ complexity theory
 - ▶ mathematical logic
 - ▶ programming
- ▶ background in Artificial Intelligence:
 - ▶ Foundations of Artificial Intelligence course (13548)
 - ▶ in particular chapters on state-space search

Gaps?

⇒ talk to us to discuss a self-study plan to catch up

Exam

- ▶ **written examination** (105 min)
- ▶ date and time: **January 27, 14:00–16:00**
- ▶ place: Biozentrum, room U1.111 (Maurice E. Müller-Saal)
- ▶ 8 ECTS credits
- ▶ admission to exam: 50% of the exercise marks
- ▶ final grade based on exam exclusively
- ▶ **no repeat exam** (except in case of illness)

Exercise Sheets

exercise sheets (homework assignments):

- ▶ solved in **groups of two or three** ($3 < 4$), submitted in ADAM
- ▶ weekly homework assignments
 - ▶ released Monday before the lecture
 - ▶ have questions or need help?
 - ↪ assistance provided in Wednesday exercises
 - ▶ not sure if you need help?
 - ↪ **start before Wednesday!**
 - ▶ due following Monday at 23:59
- ▶ mixture of theory, programming and experiments
- ▶ range from basic understanding to research-oriented

Programming Exercises

programming exercises:

- ▶ part of regular assignments
- ▶ solutions that obviously do not work: 0 marks
- ▶ work with existing C++ and Python code

Exercise Sessions

exercise sessions:

- ▶ ask questions about current assignments (and course)
- ▶ work on homework assignments
- ▶ discuss past homework assignments

Plagiarism

Plagiarism is presenting someone else's work, ideas, or words as your own, without proper attribution.

For example:

- ▶ Using someone's text without citation
- ▶ Paraphrasing too closely
- ▶ Using information from a source without attribution
- ▶ Passing off AI-generated content as your own original work

Long-term impact:

- ▶ You undermine your own learning.
- ▶ You start to lose confidence in your ability to think, write, and solve problems independently.
- ▶ Damage to academic reputation and professional consequences in future careers

Plagiarism in Exercises

- ▶ You may discuss material from the course, including the exercise assignments, with your peers.
- ▶ **But:** You have to independently write down your exercise solutions (in your team).
- ▶ AI support is acceptable to the same extent as it is acceptable from someone who is not a member of your team.

Consequences of clear plagiarism:

- ▶ 0 marks for the exercise sheet (first time)
- ▶ exclusion from exam (second time)

You are jointly responsible for your submission as a team.

Inconclusive Cases

If a tutor has serious doubts

- 1 The solution receives **0 marks initially**.
- 2 This is a **temporary assessment**, not a final accusation of plagiarism.
- 3 If you did not plagiarize, take it as a compliment. Your solution was probably exceptionally good.

How to get the marks

- 1 Come to the next exercise session.
- 2 Explain your solution and reasoning to your tutor.
- 3 If you can demonstrate that you understand the solution, **the marks are awarded**.

Exercises too difficult? We are happy to help!

Special Needs?

- ▶ We (and the university) strive for equality of students with disabilities or chronic illnesses.
- ▶ Contact the lecturers for small adaptations.
- ▶ Contact the Students Without Barriers (StoB) service point for general adaptations and disadvantage compensation.

A1.3 Course Content

Learning Objectives

Learning Objectives

- ▶ get to know theoretical and algorithmic foundations of classical planning and work on practical implementations
- ▶ understand fundamental concepts underlying modern planning algorithms and theoretical relationships that connect them
- ▶ become equipped to understand research papers and conduct projects in this area

Course Material

course material:

- ▶ slides (online)
- ▶ **no textbook**
- ▶ additional material **on request**

Git Repository

- ▶ We use a git repository for programming exercises and for demos during the lecture.
- ▶ Setting up the repository is your first task for the exercises.

TL;DR version:

Clone the Repository

```
# If you have SSH keys set up for GitHub:
$ git clone \
  git@github.com:aibasael-teaching/planopt-hs26.git
# Otherwise:
$ git clone \
  https://github.com/aibasael-teaching/planopt-hs26.git
```

Demo Examples

When working with the repository, go to its base directory:

Base Directory for Demos and Exercises

```
$ cd planopt-hs26
```

One-time demo set-up (from the base directory)
if the necessary software is installed on your machine:

Demo Set-Up

```
$ cd demo/fast-downward
$ ./build.py
```

Under Construction. . .



- ▶ Advanced courses are close to the frontiers of research and therefore constantly change.
- ▶ We are always happy about feedback, corrections and suggestions!