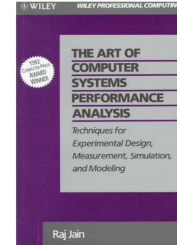


Course setup

The first 80% – compulsory

- Lectures
 - Theory, instructions, examples, Q&A
- Exam
 - Written exam with open-ended questions
- Practicum (lab)
 - 3 main assignments
 - Tools: ROS and Petri-nets
 - Required: basic knowledge of C++ and Linux



In the course, we will have two lectures per week.

A written exam at the end.

6 sessions of lab starting from next week Friday

3 main lab assignment

Course setup

The last 30% – elective

- In-lecture quizzes
- Take-home questions
- Extra lab questions
- Project

10%
bonus

Customizable part

$80\% + 30\% = 110\%$ 😊

We cannot demand presence at lectures, so you can still earn a mark 10

Go and check the fine print

Grading scheme

The fine print

Final grade = $\min(10, \frac{E + C}{100}) * M$

E = Exam

C = Customizable points

M = {pass=1 | fail=0} (mandatory assignments)

$400 \leq E \leq 800$
 $0 \leq C \leq 300$

In-lecture quizzes (5x)	20 pts each
Take-home questions (2x)	10 pts each
Extra lab questions	≤ 120 pts
Project	80 - 120 pts

TU Delft IN4390 6

Fine fine-print:

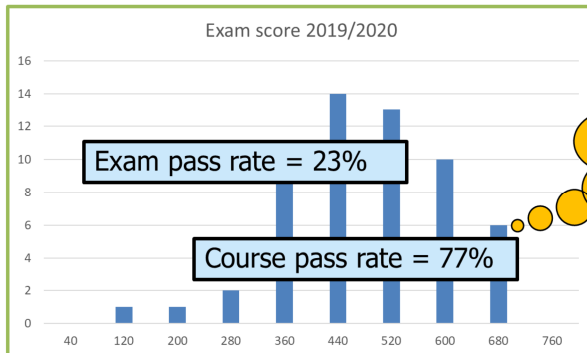
- max grade is 10 😊
- you need to score at least a mark 5 at the exam, which equals 400/800
- compulsory lab assignments may have 1 resit
- we take the 5 best quiz results (out of 9 quizzes in total)
- the points for ‘extra lab questions’ are capped at 120 (so equal to doing a project), but the total goes up to 280

Grading scheme

A word of warning

$$\text{Final grade} = \min\left(10, \frac{E + C}{100}\right) * M$$

$$400 \leq E \leq 800$$
$$0 \leq C \leq 300$$



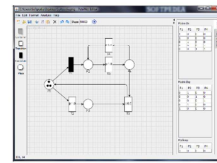
plan your customizable path now!

Mandatory assignments

With customizable extras

- Assignment 0
 - Paper reading; questions on BS
- Assignment 1 (Lab 1)
 - Measurement-based performance evaluation of ROS 2.0 communication
- Assignment 2 (Lab 2)
 - Behavior modeling and analysis using Petri-nets
- Assignment 3 (Lab 3)
 - Derive a petri-net model from a ROS application and analyze it

Pair programming



Let's talk a bit more about labs

Projects

Customizable points

- Tool demo
 - pick an existing performance/modeling tool
 - evaluate it
 - report experience (in class, as report)
- Application study
 - pick existing application/software
 - model or evaluate it
 - report experience (in class, as report)

Get approval
before starting!

We want to make sure it is doable and unique
(different teams different tools/apps)