

# 25 Years of Teaching

Gerth Stølting Brodal

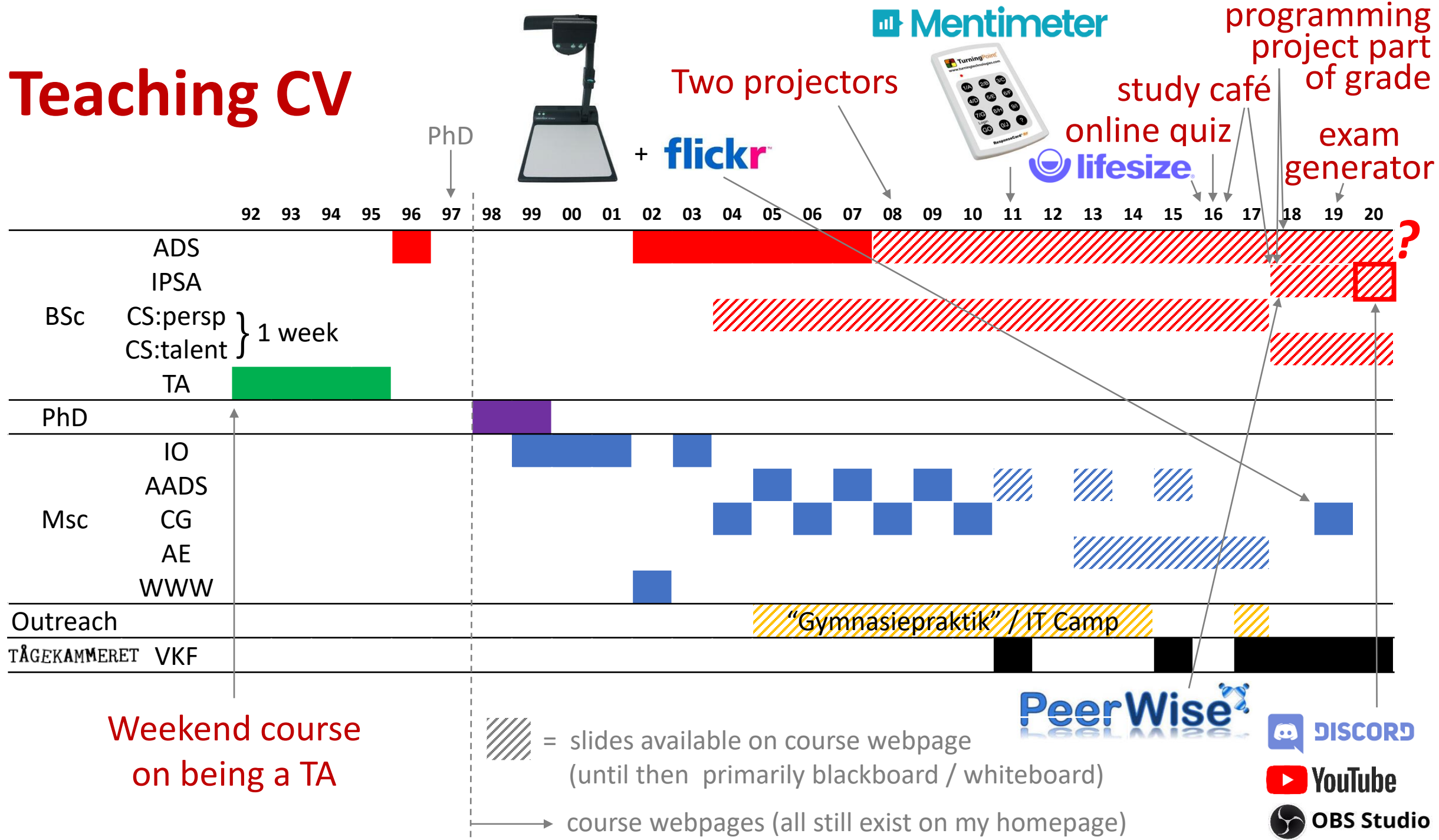
Department of Computer Science  
Aarhus University

## Curriculum Vitæ

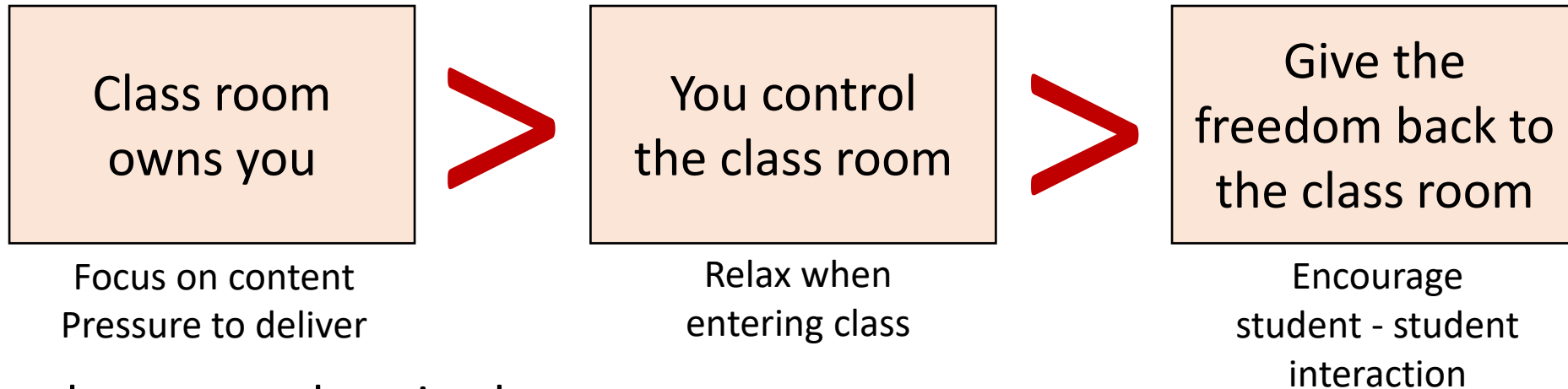
- 1989–1997 PhD Computer Science, Aarhus University
- 1997–1998 PostDoc at the Max-Planck-Institut for Computer Science, Saarbrücken, Germany
- 1998–        Research Assistant Professor / Research Associate Professor /  
Associate Professor (non-tenured / tenured (2004) / MSK), Aarhus University
- 2014–2019 Chair of the education committee Department of Computer Science




# Teaching CV



# Teaching style



- Accept that you make mistakes
- Try to be informal, relaxed, provoking, interacting (also with +150 students)
-  No shoes during lectures... [ citation from course evaluation: “Shoes are overrated” ]
- Try to start where the students are
  - a wrong solution is a good starting point for a discussion
    - Anecdote: +100 students in an auditorium tried to convince me that a (wrong) algorithm worked. I felt under heavy group pressure... After 10 min discussion I disproved the proposed algorithm.
- Students have fun  $\Rightarrow$  You have fun

# Teaching Assistants

- Should ideally take **ownership** of course
- TAs are provided with **solutions** to exercises (they are payed for helping students not for solving exercises)
- TA meetings
  - **TA-to-TA discussion**
  - going over exercises, discussing pitfalls
  - best practice in TA class
  - TAs cover more years (knowledge transfer)
- Several TAs do their own TA evaluation in TA class

## Course Management

> Content Collection →

> Course Tools

> Evaluation →

∨ **Grade Center** →

Needs Grading

Full Grade Center

Hold DV1 - Martin

Hold KE - Andreas

Hold MA1 - Josef

Hold MA2 - Jens

Hold MØ1 - Helle

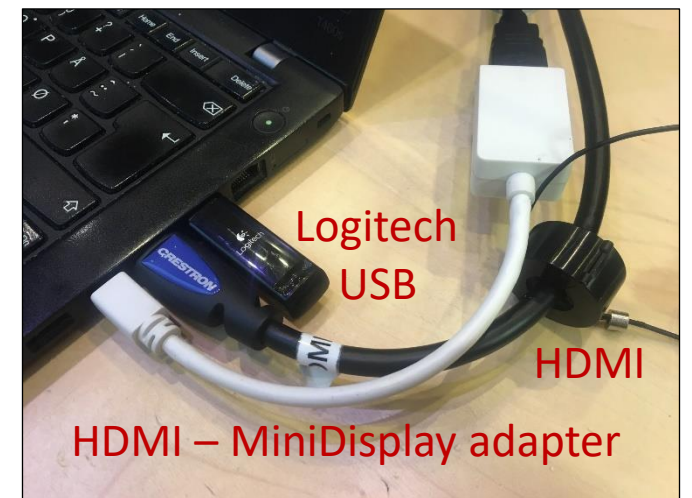
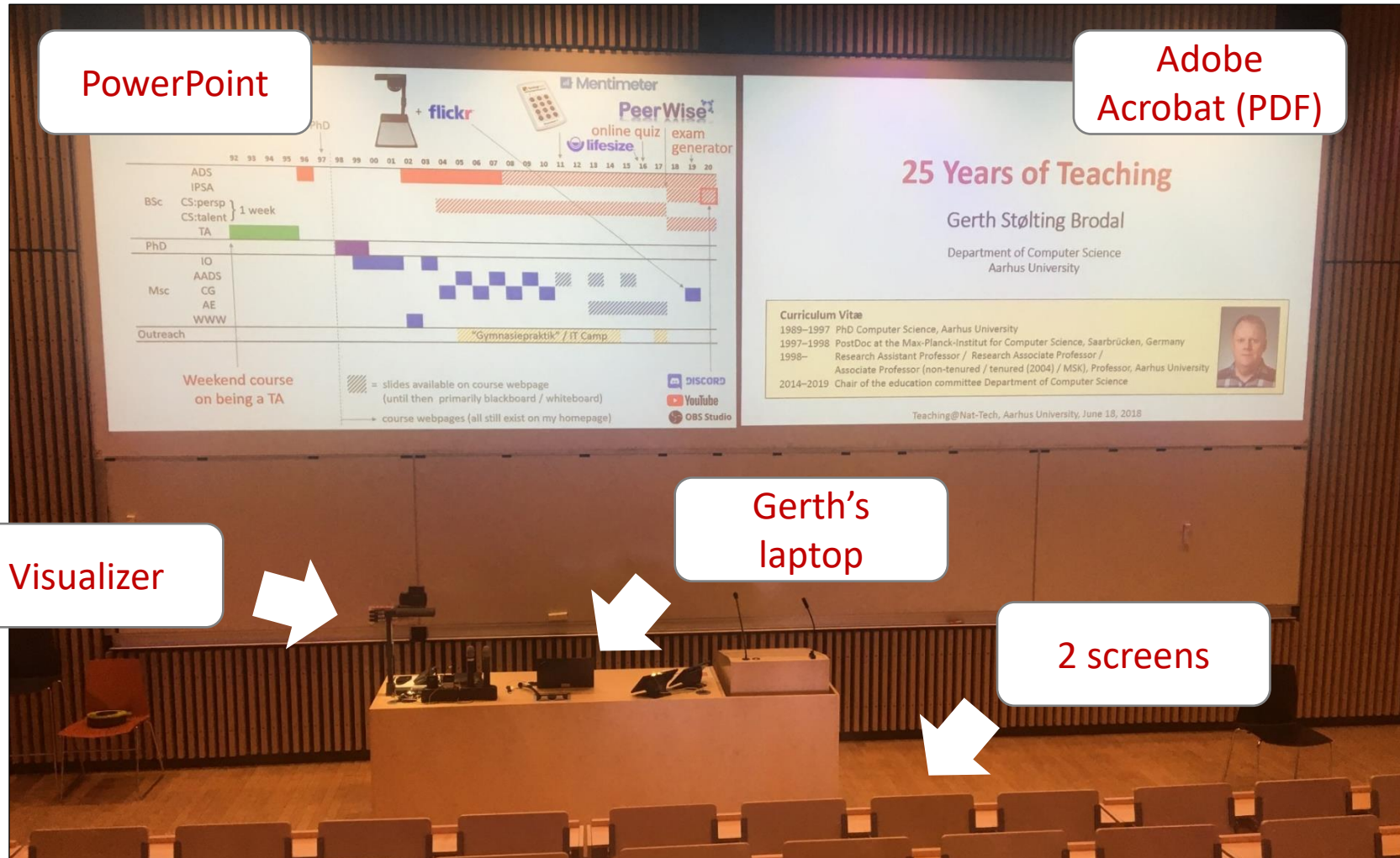
Hold MØ3 - Nikolaj

Hold TV - Niels

smartviews {

> Users and Groups

# Using two projectors



# Controlling two projectors with Logitech R700 (when you like to run around in the auditorium)



laser pointer (small red point •, hard to track)

next slide (page down)

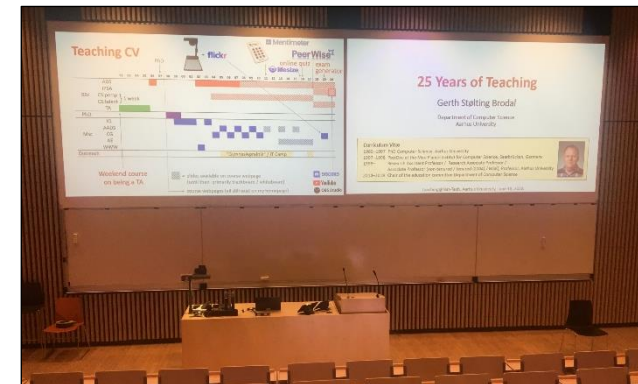
~~black screen (.)~~

reprogrammed to alternate focus between PowerPoint and Adobe Acrobat

(page up) previous slide

~~(ESC / F5)~~ start presentation

reprogrammed to send Ctrl-L in Adobe Acrobat and ESC / Shift-F5 in PowerPoint



# works also with Logitech Spotlight



hold to alternate between  
PowerPoint and Adobe Acrobat  
(when configured to “.”)


**25 Years of Teaching**

Gerth Stølting Brodal

Department of Computer Science  
Aarhus University

**Curriculum Vitæ**

- 1989–1997 PhD Computer Science, Aarhus University
- 1997–1998 PostDoc at the Max-Planck-Institut for Computer Science, Saarbrücken, Germany
- 1998– Research Assistant Professor / Research Associate Professor / Associate Professor (non-tenured / tenured (2004) / MSK), Professor, Aarhus University
- 2014–2019 Chair of the education committee Department of Computer Science



Teaching@Nat-Tech, Aarhus University, June 18, 2018

...easier to follow and gets recorded on 

## Auto Hot Key script

```
;; Map '.' to switch toggle between AcroRead and PowerPoint presentation

$.:: ; used $ to avoid recursion
if WinActive("ahk_class AcrobatSDIWindow") && WinExist("ahk_class screenClass")
    WinActivate ahk_class screenClass
else
    if WinActive("ahk_class screenClass") && WinExist("ahk_class AcrobatSDIWindow")
        WinActivate ahk_class AcrobatSDIWindow
    else
        Send . ;; Default send '.' if other window not present,
                ;; or other type of window
return

;; In AcroRead allow Logitech presenter to toggle Fullscreen
;; by mapping escape and F5 to Ctrl-L (remote alternates between
;; submitting F5 and escape)

$ESC::
if WinActive("ahk_class AcrobatSDIWindow")
    send ^l
else
    send {ESC}
return

$F5::
if WinActive("ahk_class AcrobatSDIWindow")
    send ^l
else
    if WinActive("ahk_class PPTFrameClass")
        ;; For PowerPoint send Shift-F5 instead of F5
        ;; to continue presentation at current slide
        ;; (instead of starting from slide 1)
        send +{F5}
    else
        send {F5}
return
```







**Keep it simple**  
**– for the students**  
**all pages available**  
**at semester start**

[blackboard.au.dk/webapps/blackboard/execute/courseMain?course\\_id=133812\\_1](https://blackboard.au.dk/webapps/blackboard/execute/courseMain?course_id=133812_1)

The screenshot shows the Blackboard interface for a course. At the top, the Blackboard logo and 'blackboard.au.dk' are visible. The course title 'F20 - Introduktion til programmering med videnskabelige anvendelser [520171U028]' and its English translation 'Introduction to Programming with Scientific Applications (Spring 2020)' are displayed. A sidebar on the left lists course navigation options: 'Introduction to Programming with Scientific Applications (Spring 2020)', 'Announcements', 'Course plan', 'Exercises', 'Final project', 'Handins', 'Groups & participants', 'Exam', 'PeerWise', 'Installing Python', 'Python resources', and 'AU course description'. The main content area features a 'Welcome' section with a paragraph about the course's focus on Python 3 and scientific applications, followed by a paragraph about the course structure (lectures, TA classes, and study café). Below this is a 'Course content' section with a paragraph and a bulleted list of topics: Basic programming constructs, Object orientation, Basic algorithmic techniques, Systematic development of programs, and File-based input/output.

blackboard.au.dk

Welcome to AU Blackboard

F20 - Introduktion til programmering med videnskabelige anvendelser [520171U028] Introduction to Programming with Scientific Applications (Spring 2020)

### Introduction to Programming with Scientific Applications (Spring 2020)

#### Welcome

Welcome to the course *Introduction to Programming with Scientific Applications*. The course gives an introduction to the Python 3 programming language using the book "Introduction to Computation and Programming Using Python With Application to Understanding Data" by John Guttag. The book covers the basics of Python and contains a long list of scientific applications. For the more subtle features of Python, students are encouraged to seek information online in e.g. the Python language specification.

The course will be run with 2 x 2 hours of weekly lectures, 3 hours of TA classes ("øvelser"), and 3 hours of staffed study café.

During the course students are required to hand in 10 weekly handins and one larger implementation project. Handins and the project is done in groups of up to three persons. The final exam will be a multiple-choice exam without aids, and *the final grade will be based on overall evaluation of the project and the multiple choice exam.*

#### Course content

The course gives an introduction to programming with scientific applications. Programming concepts and techniques are introduced using the Python programming language. The programming concepts are illustrated in other programming languages. The following content is included.

- Basic programming constructs: Data types, operators, variables, flow of control, conditionals, loops, functions, recursion, scope, exceptions.
- Object orientation: Abstract data types, classes, inheritance, encapsulation.
- Basic algorithmic techniques: Sorting, binary search, dynamic programming.
- Systematic development of programs: Testing and debugging.
- File-based input/output, numerical analysis, functional programming.



The screenshot shows the Blackboard interface for a course. The header includes the Blackboard logo and 'blackboard.au.dk'. The course title is 'F20 - Introduktion til programmering med videnskabelige anvendelser [520171U028]'. The page is titled 'Course plan'. A sidebar on the left contains a navigation menu with items like 'Introduction to Programming with Scientific Applications (Spring 2020)', 'Announcements', 'Course plan', 'Exercises', 'Final project', 'Handins', 'Groups & participants', 'Exam', 'PeerWise', 'Installing Python', 'Python resources', and 'AU course description'. The main content area is titled 'Course plan' and contains a paragraph of text and a table. The text mentions a textbook by John V. Guttag and a folder for code. The table lists weekly topics, dates, literature references, slides/videos, and exercises.

blackboard.au.dk

Welcome to AU Blackboard

F20 - Introduktion til programmering med videnskabelige anvendelser [520171U028] Course plan

### Course plan

Course plan

Below [G] refers to the textbook by John V. Guttag, *Introduction to Computation and Programming Using Python With Application to Understanding Data, 2nd Edition*. 472 pages. MIT Press, 2016.

Code used on slides and used for making the figures on the slides can be found in this [folder](#).

Week	Date	Topic	Litterature	Slides & Video	Exercises
5	Monday 27/1	Semester starts			
	Wednesday 29/1	<b>Introduction to Python</b>	[G] 1	<a href="#">introduction.pdf</a>	<a href="#">lecture 1</a>
	Friday 31/2	<b>Python basics</b> variables, int, float, str, type conversion, assignment, print(), help(), type() <b>Control structures</b> if-elif-else, while-break-continue, input()	[G] 2.1, 2.2, 2.4	<a href="#">basics.pdf</a> <a href="#">control.pdf</a>	<a href="#">lecture 2</a> <a href="#">handin 1</a>
6	Monday 3/2	Exercise classes start			
	Wednesday 5/2	<b>Basic operations</b> None, bool, basic operations, strings	[G] 2.3	<a href="#">operations.pdf</a>	<a href="#">lecture 3</a>
	Friday 7/2	<b>Lists</b> Syntax, operations, copy.deepcopy <b>Control structures</b>	[G] 3.2, 5.2, 5.5	<a href="#">lists.pdf</a>	<a href="#">lecture 4</a> <a href="#">handin 2</a>

course plan  
= single page with  
all necessary links

- Not mobile friendly
- Supports Ctrl-F
- Avoid fancy Bb features
- Updated as we go



## E19 - Algoritmer og datastrukturer [520191U004]

Algorithms and Data Structures (Fall 2019)

Material

Lecture videos

Course plan

Programming exercises

Handins

Groups & participants

Announcements

**Exam**

Course description

AU course evaluation

The nerdy corner

## Exam

### Previous exams

Below are the exam questions from the exams in the first year introduction to algorithms and data structure courses since 1991. A suffix of "j", "a" or "m" indicates a reexam in January, May or August, respectively. **"/s" is the solution to** the exams. **"/%"** is the answer statistics from the exam. Be aware that the curriculum has changed (slightly) over the years.

#### *Algorithms and Data Structures (ADS, Fall 2017-)*

[20m/s](#) [20/s/%](#) [19m/s](#) [18/s](#) [18m/s](#) [17](#)

#### *Algorithms and Data Structures 1 (dADS1, Spring 2004-2017)*

[17a/s](#) [17/s](#) [16a/s](#) [16/s](#) [15a/s](#) [15/s](#) [14a/s](#) [14/s](#) [13a/s](#) [13/s](#) [12a/s](#) [12/s](#) [11a/s](#) [11/s](#) [10a/s](#) [10/s](#) [09a/s](#) [09/s](#) [08a/s](#) [08/s](#) [07a/s](#) [07/s](#) [06a/s](#) [06/s](#) [05a/s](#) [05/s](#) [04a/s](#) [04/s](#)

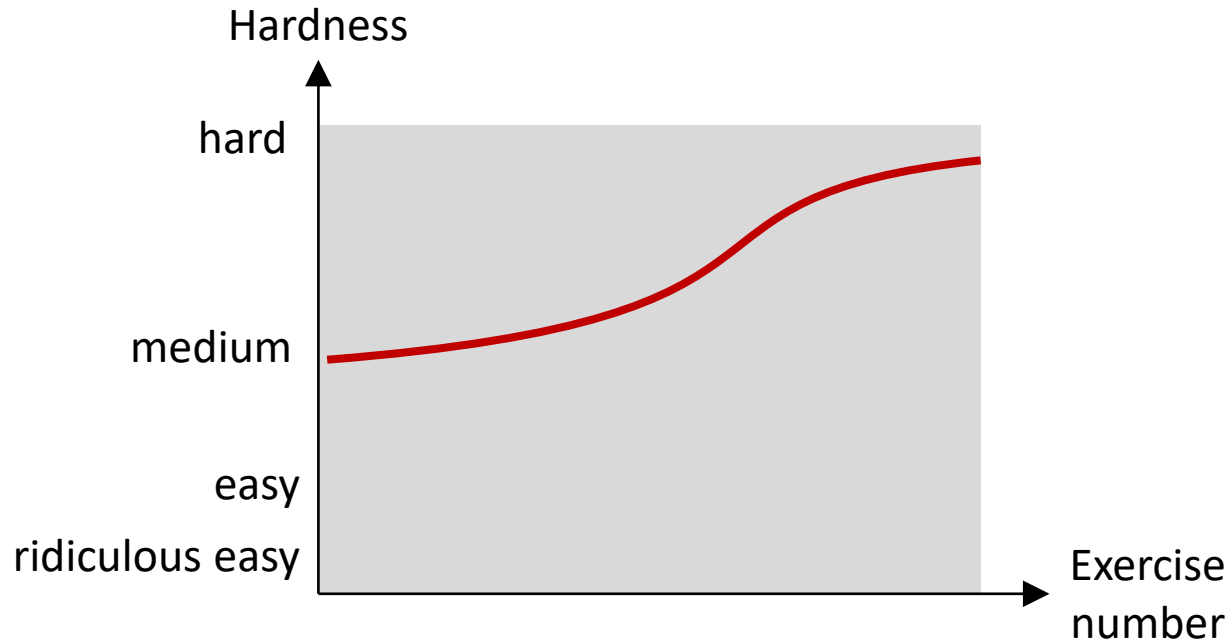
#### *Algorithms and Data Structures 2 (dADS2, Spring 2004-2017)*

[17a/s](#) [17/s](#) [16a/s](#) [16/s](#) [15a/s](#) [15/s](#) [14a/s](#) [14/s](#) [13a/s](#) [13/s](#) [12a/s](#) [12/s](#) [11a/s](#) [11/s](#) [10a](#) [10/s](#) [09a](#) [09/s](#) [08a/s](#) [08/s](#) [07a](#) [07/s](#) [06a](#) [06/s](#) [05a](#) [05/s](#) [04a](#) [04/s](#)

#### *Algorithms and Data Structures (dADS, Spring -2003)*

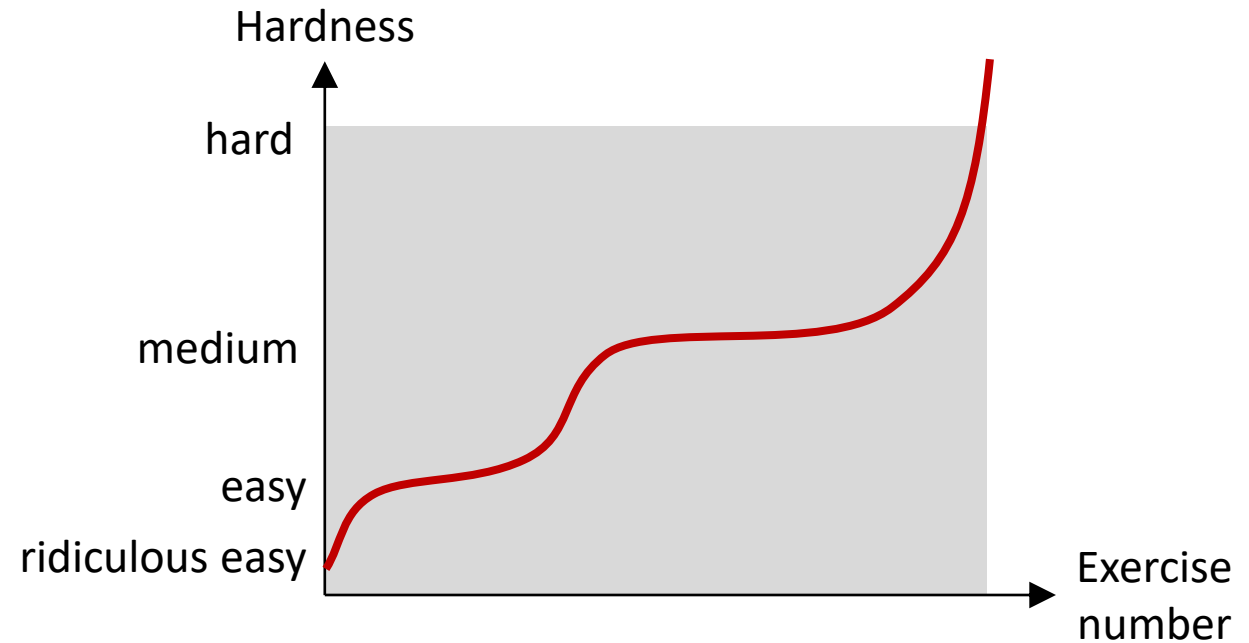
# Written Exams

*Early years*



- Too many challenging exercises
- Hard to differentiate around passing grade

*More recently*



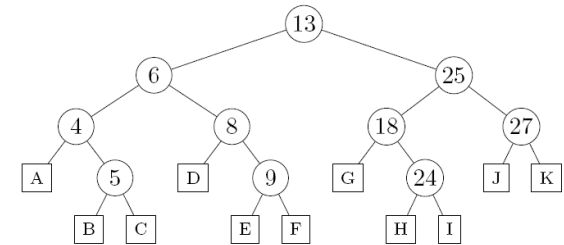
- Sufficiently many easy exercises to get evidence for passing
- Challenge top students

# Fall 2019 – Exam generator

- I have used the same (types of) questions for many years
- Made a script to generate a PDF with 126 pages of exam training questions (covering most types of exam questions)
- Students verify answers in Acrobat Reader

Grade	-3	00	02	4	7	10	12
2019	1%	4%	1%	6%	26%	26%	35%
2018	1%	4%	10%	16%	30%	27%	12%
2017	0%	6%	10%	24%	28%	19%	13%

Opgave 25 (Indsættelser i søgetræer, 4%)






Angiv i hvilke blade A–K i ovenstående ubalancerede binære søgetræ elementerne 19, 28, 21, 12 og 16 skal indsættes (det antages at før hver indsættelse indeholder træet kun ovenstående ti elementer).

	A	B	C	D	E	F	G	H	I	J	K
INSERT(19)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
INSERT(28)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
INSERT(21)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
INSERT(12)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
INSERT(16)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Python → LaTeX + tikz → PDF with embedded JavaScript

# Spring 2020 experience – going online

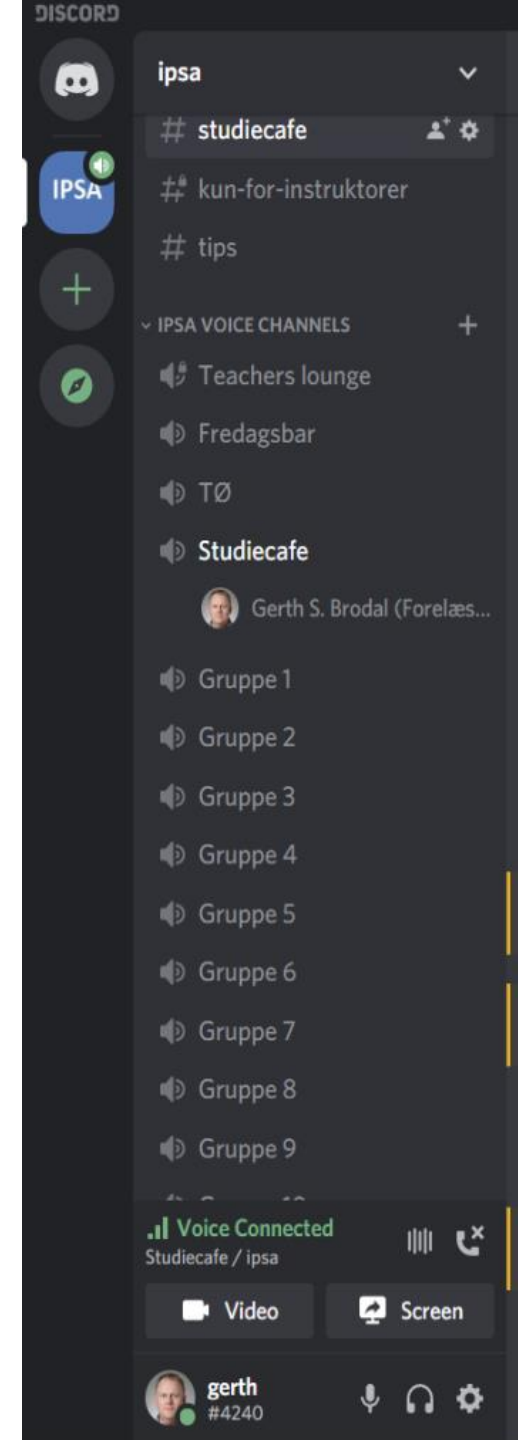
## Introduction to Programming with Scientific Applications

- March 11, 20:00 – Denmark closed down
- March 11, 23:16 – TAs suggest to move to  **DISCORD**
- March 12, 00:31 – Discord server running
- March 12, 9:15 – first TA session on Discord
- March 13, 10:15 – first study café on Discord
- Lectures, asynchronous on  **YouTube**  
(spend lot of time improving slides, technical content)  
(recorded with  **OBS Studio** )



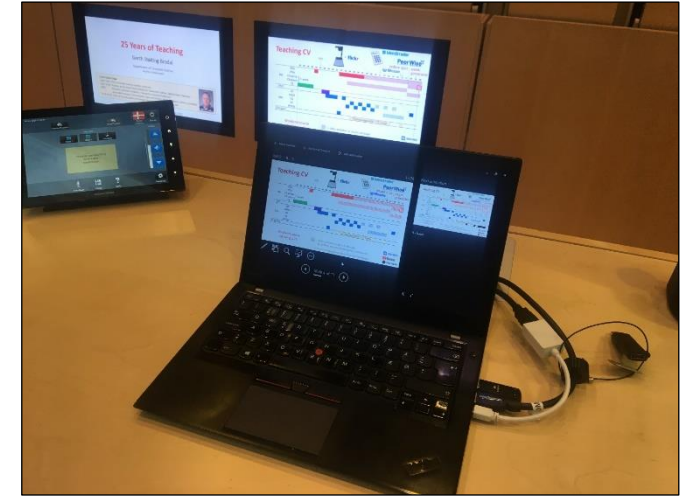
I spend  $\approx$  4 hours every week in the Discord study café interacting with students, screen sharing a lot of code...

(  discussion board completely silent !)



# Final remark

- Courses evolve over time
- Attend teaching activities offered...
  - You will likely be confirmed in a lot you already know and do, but
  - you get a change to **reflect** on your teaching,
  - **a single new minor idea can change a lot in your course,**
  - a chance to **interact** and get inspired by colleagues from other fields



Thanks to STLL