

# Concurrency: State Models & Design Patterns

*Practical Session*

***Week 01***

First things first...

# Please ensure that...

- ... you have carefully read the "PLEASE READ THIS FIRST" on the course web page.
- ... you have registered on Academia for the course.
- ... you have registered on Piazza.
- ... you always submit your solutions before the deadline.
- ... you are aware that the exercises account for 30% of your final grade.
- ... you ask questions whenever something remains unclear.
- ... you do not miss the two (mandatory) labs.
- ... you behave unlike a copycat!

**You have to attend the lecture to reveal such slides.\***

---



*\*Disclaimer:*

*The content that has been shown on this slide is irrelevant for the exam.*

# Assignment 01

***Preview***

# A01 - Exercise 1 (7 pts)

- a) **Do recent central processing units (CPUs) of desktop PCs support concurrency?  
Why became concurrency for many software applications very important these days?**
- b) **Why do we need synchronization mechanisms in concurrent programs?**
- c) **What is safety? Give one concrete example of a safety violation.**
- d) **What is liveness? Give a concrete example of a liveness violation.**
- e) **Can a binary semaphore lead to a deadlock? Why?  
Can it lead to starvation? Why?**
- f) **How do monitors differ from semaphores? Please provide a precise answer.**
- g) **How are monitors and message passing similar?  
How are they different?**

## A01 - Exercise 2 (2 pts)

$x := 1$

Thread 1  $\rightarrow x := x + 7.$

Thread 2  $\rightarrow x := x * 5.$

**Considering the code above: Give all possible values of  $x$  at the end of the execution of both threads with their corresponding execution traces.**

# A01 - Exercise 3 (1 pt)

**Implement a monitor using a binary semaphore.  
Use pseudo-code and comment it.**