



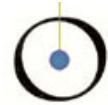
Concurrent Object-Oriented Programming

Bertrand Meyer, Piotr Nienaltowski

Lecture 1: Course overview and introduction

Updated: 04 April 2006

Chair of Software Engineering



Practical details

Schedule:

Course: Tuesdays 14-16, IFW D42

Exercises: Tuesdays 13-14, IFW D42

Course page (check it at least once a week):

<http://se.inf.ethz.ch/teaching/ss2006/0268>

Assistant:

Piotr Nienaltowski

Contact:

- concur-course@se.inf.ethz.ch
- piotr.nienaltowski@inf.ethz.ch
 - 044 632 44 68



Grading



Exam 35%

Will be held at the end of the semester
(**not** in the semester break)

Exam date (to be confirmed): lecture time on 27 June

Project 65%

Application to be built using *SCOOP*





The purpose of this course

- To give you a practical grasp of the excitement and difficulties of building modern concurrent applications
- To expose you to newer forms of concurrency
- To study how the object-oriented paradigm transposes to concurrent settings, and how it can help address concurrency issues
- To introduce you to the main concurrent O-O approaches and give you an idea of their strength and weaknesses
- To study in depth one particular approach: SCOOP
- To enable you to get a concrete grasp of the issues and solutions through a course project



Two Sides of the Same Coin

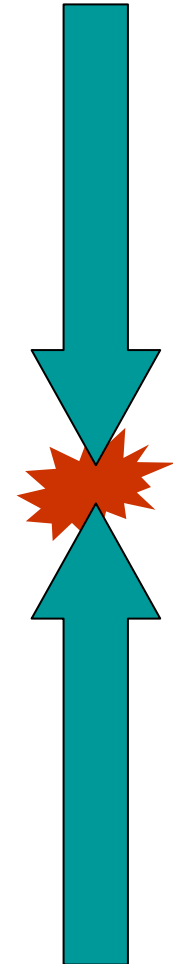


SCOOP part

- The "object lesson"
- High-level support for concurrency
- Concurrency solution integrated with an OO programming language, i.e., Eiffel

"Classic" part

- The "old school" approach
- Following historical evolution
- Problem/solution, illustrations, e.g., Java





Course description (from catalog)

This course explores the connections between the object oriented and concurrent programming paradigms, discussing the problems that arise in the process of attempting to merge them.

It reviews the main existing approaches to concurrent O-O computation, including both widely used libraries for multi-threading in Java and .NET and more theoretical frameworks, with a particular emphasis on the SCOOP model.

It also provides some of the formal background for discussing the correctness of concurrent O-O applications.



What you should do by next lecture



Read SCOOP introduction at

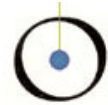
http://se.inf.ethz.ch/people/nienaltowski/papers/scoop_easy_draft.pdf

and start reading the concurrency chapter at

<http://www.inf.ethz.ch/~meyer/publications/oosc/concur.pdf>

(or in *Object-Oriented Software Construction*, 2nd edition,
available from Polybuchhandlung)





End of lecture 1

