Lesson 0

Concurrent Programming
Rinnakkaisohjelmointi (RIO)

581332

Teemu Kerola

University of Helsinki

Department of

Computer Science

Autumn 2009

http://blueballfixed.ytmnd.com/

Concurrent Programming Autumn 2009

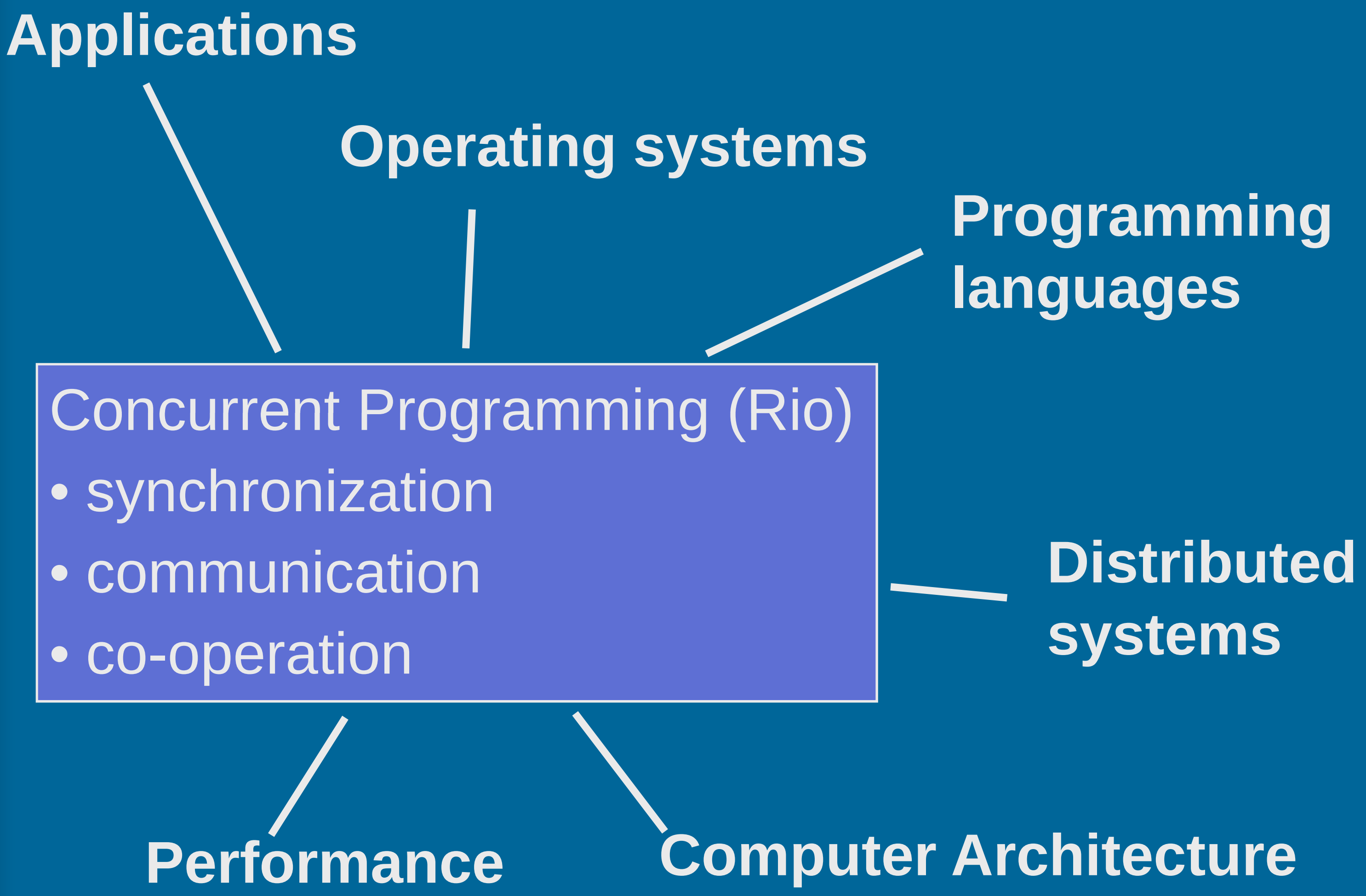
- Administration

  Lecture 0

  - Course area and learning goals
  - Learning methods
- Introduction to Concurrency

  Lecture 1
- Slides in English, lectures in Finnish
  - Non-Finnish speakers can read the slides even if they do not understand the lectures

Connections



Applications

Operating systems

Programming languages

Concurrent Programming (Rio)

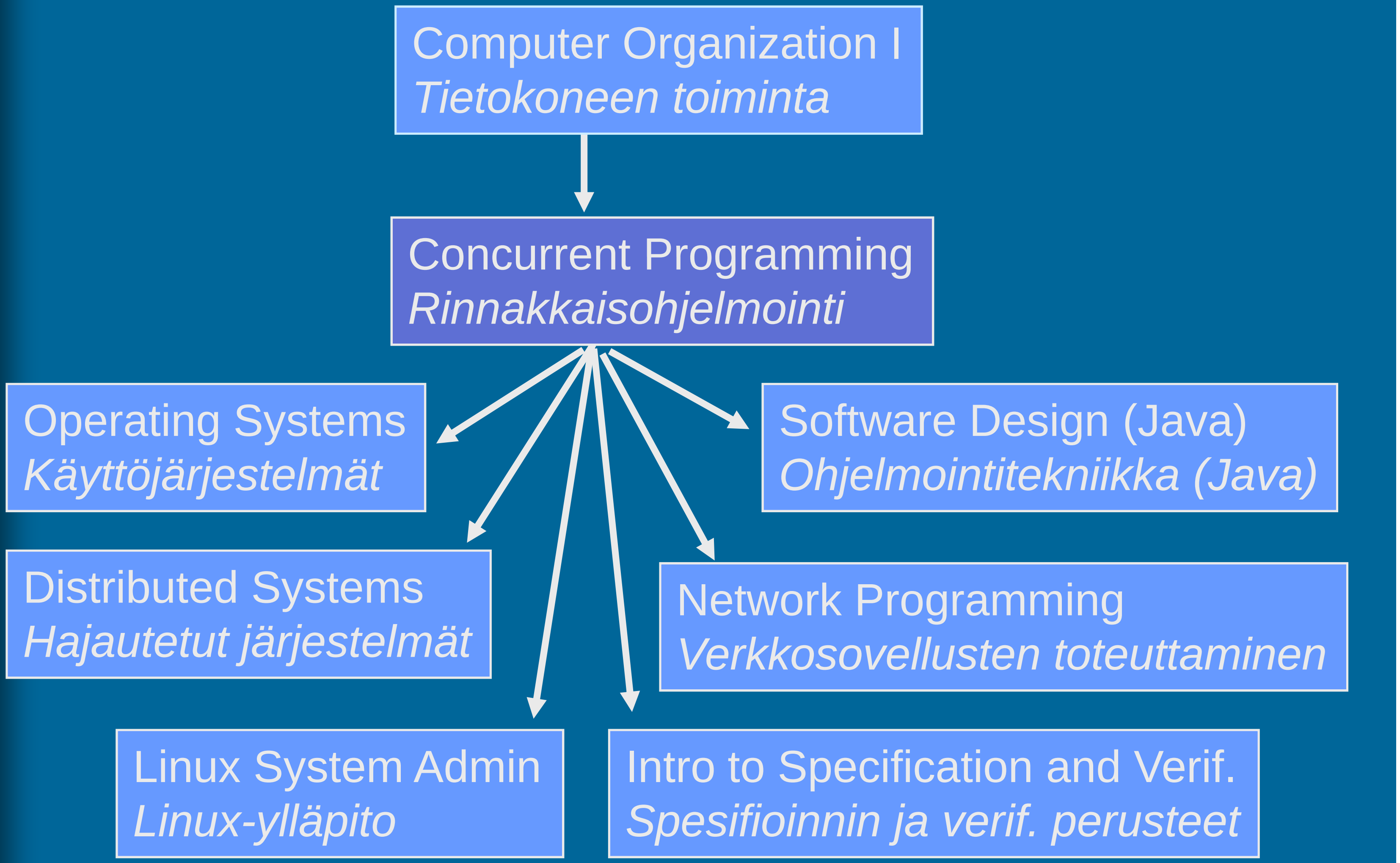
- synchronization
- communication
- co-operation

Distributed systems

Performance

Computer Architecture

Course Connections



Computer Organization I
Tietokoneen toiminta

Concurrent Programming
Rinnakkaisohjelmointi

Operating Systems
Käyttöjärjestelmät

Software Design (Java)
Ohjelmointitekniikka (Java)

Distributed Systems
Hajautetut järjestelmät

Network Programming
Verkkosovellusten toteuttaminen

Linux System Admin
Linux-ylläpito

Intro to Specification and Verif.
Spesifioinnin ja verif. perusteet

Motivation

- To know and understand …
  - Why concurrency is important
  - What are the concurrency problems in your systems
  - How concurrency problems are usually solved
  - What are the usual tools for solving concurrency problems
  - How concurrency problems may be solved at different system levels
    - HW, OS, progr. language library, application

Learning Goals

- Concurrency and problems caused by concurrency
  - Basics, fundamental ideas, background
- Fundamental concepts and models in concurrency
  - Main emphasis is here!
  - E.g., “can explain special features of semaphores and monitors and use them properly in applications”, or “can explain how deadlocks can be prevented”.
- Concurrent programming in distributed systems
  - Get taste of it, more in later courses
- Concurrent programming in practice
  - Discussed in lectures, practice in project

What Good is Concurrent Programming for?

- All computer systems are inherently concurrent – one must understand concurrency to understand computer system operation
- Concurrency allows <u>huge speedups</u> for properly designed systems
- Concurrency causes <u>complex problems</u> that are not easy to solve without good understanding of concurrency

- Should I use threads in my Java application or not?
- **Would it be better to use locks, semaphores, monitors or transactional memory to solve synchr. and communication problems in my application?**
- Why doesn't it do what I thought it would do?
- How can I show my boss that it really works?

What is Not Covered?

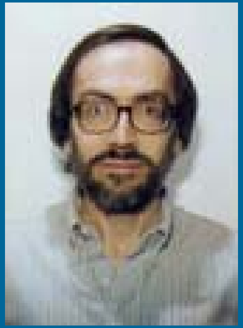
- How to write efficient code for multicore architectures?
- What types of applications are suitable for multicore architectures
- What programming paradigms exist to write efficient code for multicore architectures?
- How to write multicore code so that it would run in many multicore architectures?
- How to best utilize multicore GPU in your application?
  - CUDA - Compute Unified Device Architecture
  - How to partition your solution to multicore CPU & GPU?
- Programming distributed applications

Learning Methods

- Text book: Ben-Ari 2006 (& Stallings 2005)
- Lecture notes
  - Not perfect for self study – use with text book
- Summary lectures & discussions
- BACI – Ben-Ari Concurrency Interpreter
- Practice problems kertaustehtävät
- Homeworks kotitehtävät
- Project
- Group meetings (practice sessions) laskuharjoitukset
- Learning diary (elective)
- Course exam

study circle teamwork opintopiirit

Mordechai Ben-Ari
http://stwww.weizmann.ac.il/g-cs/benari/



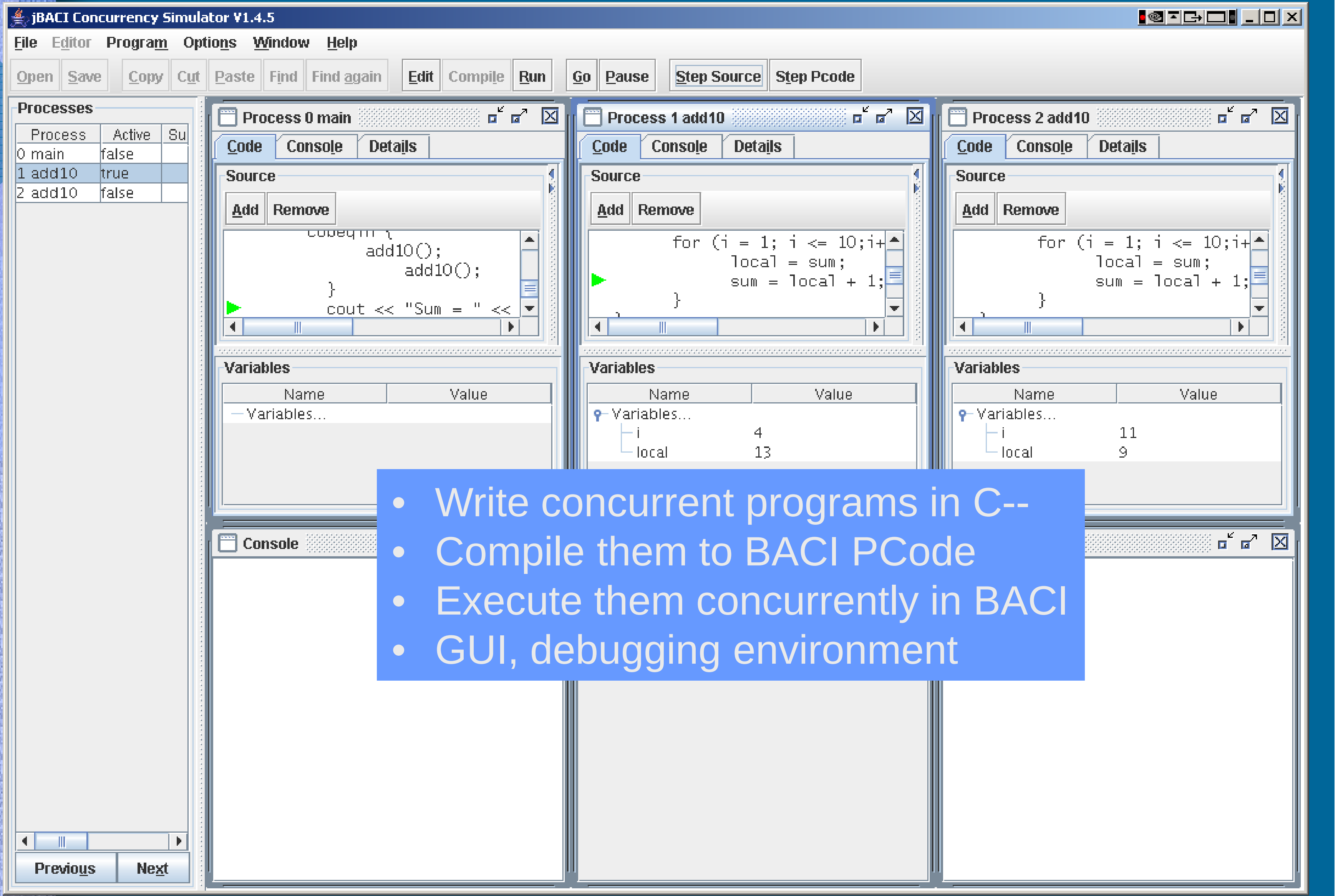
William Stallings
http://www.acm.org/crossroads/xrds10-4/stallings.html

Summary Lecture and Discussion

- Short summary lecture
  - Some of the slides for 2008 lectures
  - Assume: students have read the text book in advance
    - Basic knowledge of todays topic
- Discussion (*Socratic* discussion)
  - Goal: deeper understanding of todays topic
  - Students discuss
    - Starting points: questions, claims
      - Students, moderator
    - Goal: answer the question, verify the truth of the claim
  - Moderator may intervene to keep discussion on track

Nina Aremo (Faculty of Science) will attend some lectures and may make a short interview study on how well this learning method works for this course

BACI – Ben-Ari Concurrency Interpreter



jBACI Concurrency Simulator V1.4.5
File Editor Program Options Window Help
Open Save Copy Cut Paste Find Find again Edit Compile Run Go Pause Step Source Step Pcode
Processes
Process 0 main
Process 1 add10
Process 2 add10
Code Console Details
Source
Add Remove
Variables
Console
Previous Next

Process	Active	Su
0 main	false	
1 add10	true	
2 add10	false	

Process 0 main

```
    cobegin {
          add10();
              add10();
        }
        cout << "Sum = " <<
```

Name	Value
Variables...	

Process 1 add10

```
    for (i = 1; i <= 10;i+
          local = sum;
          sum = local + 1;
    }
```

Name	Value
Variables...	
i	4
local	13

Process 2 add10

```
    for (i = 1; i <= 10;i+
          local = sum;
          sum = local + 1;
    }
```

Name	Value
Variables...	
i	11
local	9

- Write concurrent programs in C--
- Compile them to BACI PCode
- Execute them concurrently in BACI
- GUI, debugging environment

Java

- How to use Java for Concurrent Programming?
  - Threads
  - Messages
  - Semaphores
  - Monitors
- Lectures, projects

säikeet

viestit

semafori

monitori

Study Circles

opintopiirit

- Group work, team work
  - It is better to study in a team than alone
  - Peer student support
  - Study circles formed in the first group meeting
- Student centered learning
  - The student has responsibility on learning
  - Instructor facilitated learning
  - Instructors give good environment for learning
- Three types of team work
  - Solving homework problems independently and then discussing them in study circles and in group meetings
  - Projects
  - Any other co-operative work for this course

Creation of Study Circles

- Possibly the largest problem in study circle courses
  - ”Ville promised, but did not do and he was not accessible. And then Maija did most of it. This is not right! Boohoo! ☹”
- Study circles are formed in the 1st group meeting
  - Goals should be similar
    - Easy in real life: ”you will finish it or …”
  - Think about your goals before the 1st group meeting
    - Do I want to learn a lot, or just pass the course?
    - Will I help others, or just concentrate on my own work?
  - Discuss and agree on common goals before agreeing on forming a study circle
    - Finally, sign the ”Study Circle Contract”

http://www.cs.helsinki.fi/u/kerola/rio/opintopiiri_sopimus.html

- Keep up with your agreement
  - Inform the study circle immediately, if you will not continue
  - Get quickly rid of peer students who do not work as agreed on

Practice Problems

- Practice problems

kertaustehtävät

  - Self evaluation
    - Do them only after you think you know the material
    - Do I understand it now?
    - They check only some part of the material, no guarantees!
  - Use does not directly affect your grade
    - No bookkeeping on material use
    - No credit toward course grade
  - More practice problems are made in the project

Homework Problems

- Normal homework problems
- Learning happens when you *solve* the problems and *discuss* them
  - Study topic area first before trying out the problems
  - Work on the problems independently before discussing them
  - Reading a complete solution or giving one to peer student is wasting a good problem!
- Homework problems are discussed at
  - Study circle own meetings before group meetings
  - Group meetings with peer students at the table
- Affects your grade
  - You get homework points (hwp, lhp) for completed problems
  - Only for those present in group meeting

http://www.cs.helsinki.fi/u/kerola/rio/s2009/laskuharj/lh1.html

Group Meetings

Laskuharjoitus

- Mark down the problems you have completed
- Assistant will organize you in tables
  - Each table should have a student solution for each problem
  - One or more tables may be in English in English speaking practice session
- Discuss all problems in your own table
  - You should have at least tried to solve all problems beforehand
  - You should understand all solutions at end
  - Other students in your table should understand all solutions at end
  - Correct solutions available, consult assistant if needed
  - No presentations in front of class
- Discuss additional topics given in solutions paper
  - Open ended discussions, no ”correct” answers given
- Common discussion of selected problems/solutions
- Advice students in other tables, if there is time for it

Project 1

http://www.cs.helsinki.fi/u/kerola/rio/s2009/projektit/pr1.html

- Project (A&B or A&C) with team work
  - A. Come up with a new practice problems
  - B. Do a small project in Java from text book
  - C. Write a better guide on how concurrent programming is done with Java

    http://www.cs.helsinki.fi/u/kerola/rio/ohjeet/ohjeet.html
- Goal is deeper understanding on some topics
  - Other course components may be needed as background knowledge
- Affects your grade
  - Instructor evaluates the report
  - You get project points (pp) in three parts
    - Basic points (3 pp) for just completing the project (for 8 pp project)
    - Grade (1-5 pp) depending on the quality of work
    - Participation points (max ±2 pp) based on you participation
      - Study circle determines this part! ⇦

Learning Diary (Extra Project 2)

oppimispäiväkirja

http://www.cs.helsinki.fi/u/kerola/rio/s2009/projektit/pr2.html

- Evaluate and reflect
  - Cover all learning events (lecture, homework, practice meeting, study circle meeting, etc.)
  - What did you do, observe, learn, and feel
- Affects your grade
  - Same way as other projects
  - Points (pp) are completely extra, and you can get an excellent grade also without this project
  - With learning diary you probably get a better grade!

Studying for This Course

- Study weekly topics
  read
  - Read the text book the same topics, with different approach
    - Use lecture slides if you find them useful
  hear discuss
  - Attend summary lectures and actively participate discussions
  think
- Check your learning with self evaluation
  - Do practice problems and homework do, evaluate
- Participate in study circle
  discuss
  - Discuss homework
  discuss
  - Weekly group meeting
  do
  - Continue projects
  discuss
  - Study circle meet face-to-face or in the web
- Finish project 1 in time
  read
- Study for exam
  evaluate
- Take course exam
  do
  - Do extra project think, do, reflect

Evaluation

- Self evaluation
  - Do practice problems after each topic
    - Does not affect your grade
  - Do homeworks each week
    - Do I understand or not?
    - What is there still to learn and how do I do it??
    - Affects your grade
- Course exam
  - Gives a fixed deadline for learning
  - Covers all topics
    - Topics learned in independent study as well as in study circles using various learning methods
  - Evaluates learning
    - Most of the grade based on this
    - Must reach certain level (50%) to pass the course

Grading

- Good work is awarded
- Diligence and knowledge is awarded
- Course component maximum grade points

Homeworks (min 1p)	6 p
Study circle project (min 1p) (extra Project 2: study diary)	10 p (+5 p)
Course exam (min 22 p)	44 p
Total (min 30) (with project 2)	60 p (65 p)

Course contents

- Lecture 0: Admin
- Lecture 1: Concurrency
- Lecture 2: Concurrency at Progr. Lang. Level
- Lecture 3: Critical Section Problem
- Lecture 4: Verifying Concurrent Programs
- Lecture 5: Deadlocks
- Lecture 6: OS Support for Conc: Semaphores
- Lecture 7: More on semaphores
- Lecture 8: Progr. Lang. Support for Conc: Monitors
- Lecture 9: Concurrency Control in Distr. Environment
- Lecture 10: Crit. Sections in Distributed Environment
- Lecture 11: Practical Examples on Concurrency Control
- Lecture 12: Current Research, Course Summary
- Project: Java programming

http://www.cs.helsinki.fi/u/kerola/rio/s2009/aikataulu.html

Concurrency and problems caused by concurrency

Fundamental concepts and models in concurrency, Concurrent programming in practice

Conc. progr. in distr. systems

Conc. progr. in practice

Motto

- "It is not good exercise, if you do not sweat"
- This is not a marathon!
- Study-week approach
  - Altogether some 12 h / week
    + exam preparation + exam
  - Total some 80 h / 2 sw course (2 work weeks)

- Top-down approach

5 yrs / 300 cu = 1 yr / 60 cu = 1600 h / 60 cu
= 26.67 h / 1 cu = 107 h / 4 cu

  - Total some 107 h / 4 cu course

Nina Aremo study 2007

- Questionare study, weekly basis
  - 40/121 student participated (14 women, 27 CS majors)
- Course workload opinion: 50% ok, 50% too much
  - Aver weekly work hours: 7, 8, 10, 9, 10, 11 (compare to 12)
  - Total work hours: most 50-70, max 136 (compare to 80 or 107)
- Comments
  - "Need model solutions to homework problems"
  - "Lecturer did not take questions"
  - "Difficult to do well, if you do not attend lectures and practice sessions"
  - "Web pages are incoherent"
  - "Study circles would need more guidance"
  - "Got real busy at end because of so many other courses"

Summary

- Course administration
- Course components and learning methods
- BACI simulator
- Lecture format
- Projects