Database Systems: Administrative Notes

Prof. Dr. Jens Dittrich

1 What is a Flipped Classroom?

This lecture is organized as a *Flipped Classroom* (aka Flip Teaching), i.e. the lecturer does **not** present the teaching content and its associated materials in front of the students in the lecture hall — as it would be the case with a traditional lecture. Instead, we offer learning material for self study. This includes instructional videos (mostly self-produced), a book summarizing and extending the videos, slides, research papers, and other sources. We expect students to study this material at their own pace and prepare themselves for the weekly meetings (the so-called *LAB*). This time, we will offer two types of LABs, that take place at the time the traditional lecture would take place:

1. The *exercise LAB* (Thursday from 10:15–12:00). The purpose of the exercise LAB is to jointly apply the material, *deepen your understanding*, and prepare you for the exams. In the exercise LAB we will clarify questions and start working on the weekly *exercises* in groups of 2–3 students. The professor, the tutor in chief, several Ph.D. students, student tutor(s) will help you with that.

2. The *programming LAB* (Tuesday from 14:15–16:00, not every week, see CMS calendar). The purpose of the programming LAB is to start implementing the programming tasks (see Section 4) in groups of 2–3 students. In this session we clarify questions about the given problems and help you getting in the tasks. Again, the professor and the tutors will help you with that. Feel free to bring a laptop.

Example sequence of events for learning a given topic:

When?	**What?**
until week N, Thursday 10:15	*Self-paced:* learn the material (videos, slides, book)
week N, Tuesday 14:15 – 16:00	*Programming LAB:* start implementing programming tasks
week N, Thursday 10:15 – 10:25	*Compulsory Mini-Test:* 10 minute, one page, test before each *LAB*
week N, Thursday 10:25 – 12:00	*Exercise LAB:* start solving weekly exercises with professor's and tutor's help
week N+1, Thursday 10:00	*Optional:* submit your solution to the exercises electronically
week N+2, Tue, Wed	*Tutorial:* discuss solutions of exercises
week N+2	*Sample Solutions:* published

2 Requirements for Passing:

What:	**When:**	**Compulsory:**	**Weight:**	**To pass:**
Exercises	Every week	no	none	none
Mini-Tests (10min)	Every week	yes	none	$\geq$ 50% on average at most 2 with 0 points
Programming Tasks	Every 3–4 week(s)	yes	30%	$\geq$ 50% on average
Midterm	Dec 6, 11:45am	yes	20%	none
Final exam **or:** Repetition exam	Feb 7, 11:30am March 22, 10:00am	yes	50% 50%	$\geq$ 50% $\geq$ 50%

Midterm, Final Exam, Repetition Exam:

(a) 120 minutes each.

(b) The midterm covers material treated in the lecture until that point in time.

(c) The final exam covers all material treated in this course.

(d) **The final** exam may be repeated at the end of the exam period.

(e) You can take two two-sided physical A4-pages of your personal *handwritten* summary of the lecture content to the midterm and all exams (no print-outs, no exam question sheets, no carbon copies).

3 Learning Resources and Tools:

Book: Prof. Dittrich recently wrote a "flipped textbook". In that book links to all learning material you need (including learning goals, videos, slides, encyclopedia, and research papers) are provided. In addition, textual summaries of the videos in Q&A-style are provided. You will receive a **free electronic copy** of the book for self-study. Notice that this is copyrighted material and must not be redistributed in any way, thanks! Only few additions to this book will be made throughout the semester (if at all). Feedback and suggestions are welcome!

CMS:

(1) Log in to CMS `https://teaching.infosys.uni-saarland.de` using your student account, and check out the calendar to not miss anything.

(2) Hand-in your solutions to the exercises (optional). Only via CMS, PDF only, max. 2 MB, scans of hand-writing allowed, yet digital content preferred.

Exercise LAB / Programming LAB:

(1) Replaces the frontal lecture.

(2) Apply the material you learned, deepen your understanding.

(3) Work in groups of 2 to 3 students, *start solving the weekly exercises respectively the programming tasks.*

(4) Get supervision from the professor and the tutors.

Tutorial:

(1) Choose your preferred tutorial group in CMS. First come — first served.

(2) Discuss solutions to weekly exercises in more detail.

Forum:

(1) Discuss questions in our forum: `http://forum.infosys.uni-saarland.de`.

(2) Check out this tour: `http://stackoverflow.com/tour` if you are unfamiliar with Q&A forums.

(3) Contribute: post answers, add comments, vote up or down, accept the best answer!

4 Programming Tasks

There will be **four** programming tasks, that are developed in a single C++ codebase. For these programming tasks, we may provide interfaces and automated tests (functional and performance). Your test results plus a code inspection by a tutor determine the points for a particular programming task. In the programming LAB (on average every two weeks on Tuesday 14:15-16:00, not every week, see CMS calendar), we will start implementing the tasks, discuss problems, and answer questions.

5 Fraud / Copy

Any form of fraud will result in grading source and sink with zero points. Any attempt to present work done by others as one's own performance is rated as fraud and may result in the student losing the right to examination concerning this lecture.