



**Challenges:  
Continuous (Peer) Assessment in der Informatik-Lehre**

Manfred Meyer – Westfälische Hochschule, Bocholt, Germany

HFDcon 22 – 30. Juni 2022



## ■ Manfred Meyer

- M.Sc., Ph.D. Informatik, Uni Kaiserslautern
- Professor für BWL, insb. Wirtschaftsinformatik, FB Maschinenbau, Westf. Hochschule

## ■ Lehre im Studiengang B.Sc. Wirtschaftsinformatik

- Grundlagen der Informatik/Programmierung 1 und 2 (1. bzw. 2. Sem.)
  - Vorlesung 3 SWS
  - Praktika 3 SWS in Gruppen à ca. 15 Teilnehmer
- Algorithmen und Datenstrukturen (2. bzw. 3. Sem.)

## ■ Früher:

- Praktika am PC im Informatik-Labor: Bearbeitung von Programmieraufgaben in Java
- Prüfung (Klausur) auf Klausurpapier
  - Theorie-Fragen gemischt mit Programmieraufgaben (verschiedene Taxonomiestufen)
  - „Handschriftliches Programmieren“... ☹️

## ■ Vorlesung

- Learning Outcomes einheitlich auf Taxonomie-Stufe 5 („Synthese“ n. Bloom)
- Beachtung der „Taxonomischen Eindeutigkeit“ (Rolle von Bonuspunkten??? ☹)
- 2019 Digi-Fellowship „*Social Learning & Just-in-Time Teaching in der Informatik-Lehre*“
- Einsatz der Social-Learning Plattform „Perusall“ (Eric Mazur et.al. / Harvard)
- Fokussierung auf „Problemthemen“ (Nutzung Peer Instruction, Flipped Classroom etc.)

## ■ Praktikum

- Zunehmend weitere Sprachfeatures, Komplexität, aber immer konkrete Aufgaben
- ZQVM-Antrag „4P“ (Praxisnahe Prüfungen und Projektorientierte Praktikumsumgebung)
- Umbau des Informatik-Labors für Gruppenarbeiten („Inseln“)
- Bearbeitung praktischer Programmieraufgaben auf eigenen Notebooks (BYOD)
- Notebooks auf Leihbasis verfügbar bei Bedarf



# Kompetenzorientierte Lehre: Praktika im Labor

Studierende arbeiten in Gruppen, 27" Monitor je Gruppe, Präsentation via Beamer



## ■ Kompetenzorientierte Prüfungen im Labor:

- Prüfung im Labor unter möglichst identischen Bedingungen wie Praktika (technisch und inhaltlich → „constructive alignment“)
- Beschaffung von 30 identischen Prüfungsrechnern (Notebooks)
- Nutzung der IDE (Eclipse/Java) analog zum Praktikum, API Dokumentation etc. lokal verfügbar, kein Internet-Zugang
- Workspace auf USB-Medium
- Prüfungsplätze analog zu Praktika – mit Sichtschutzwänden zwischen den Plätzen

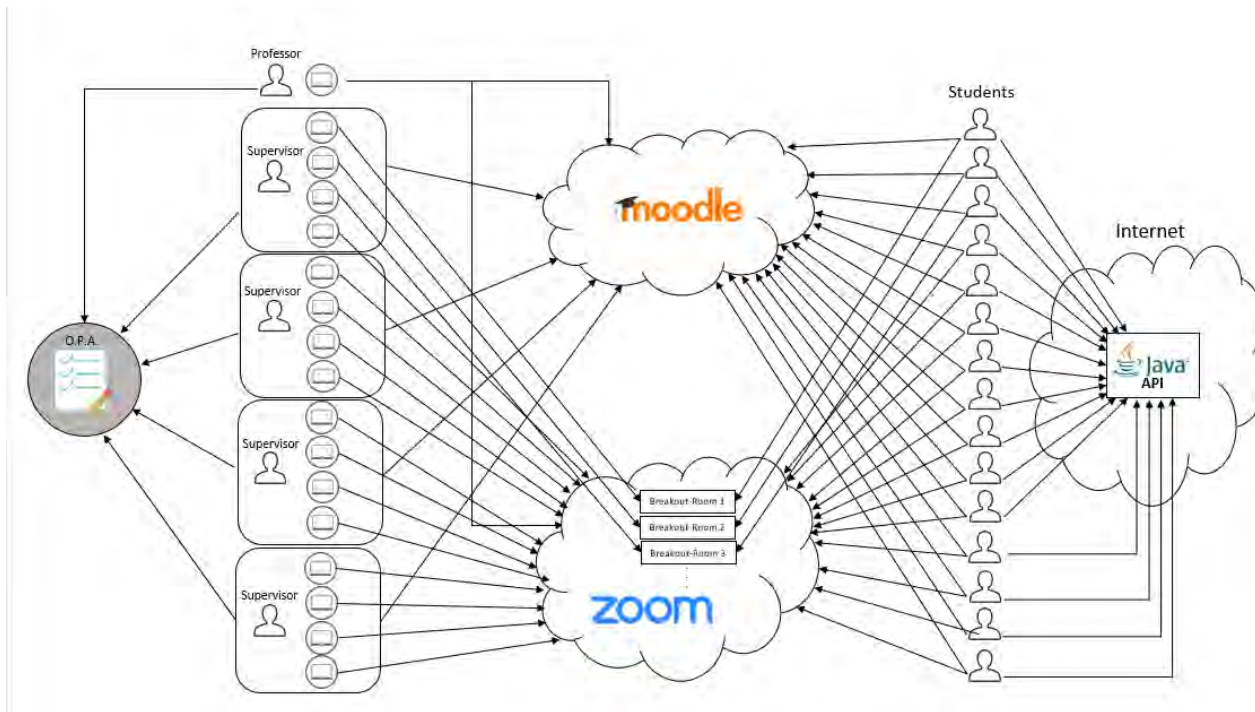




# Und dann kam COVID... und Prüfungen gehen online!

## ■ Ad-hoc Transformation nach Online unter Wahrung der Ziele:

- Prüfungen finden online von zuhause am eigenen PC/Notebook mit IDE/Java, Moodle & Zoom statt:



- **Ad-hoc Transformation vom Labor nach Online unter Wahrung der Ziele *Constructive Alignment, Fairness, Validität, Robustheit***

Paper EDULEARN 21

- **Beitrag zu „Digitalen Prüfungen mit Drittapplikationen“**

HFD Whitepaper Nr. 62,  
S. 76f.



## DON'T LET COVID-19 RUIN OUR COMPETENCY-ORIENTED EXAMS: LESSONS LEARNED FROM AN AD-HOC TRANSFORMATION OF PRACTICAL PROGRAMMING EXAMS FROM LAB TO ONLINE

Manfred Meyer

Westfälische Hochschule – Westphalian University of Applied Sciences  
Bocholt (GERMANY)

### Abstract

Competency-oriented exams offer a wide range of advantages, especially where the use and mastery of third-party applications and tools play an important role. Therefore, we developed a competency-oriented setup for both our programming classes and exams ensuring their constructive alignment. Exams were moved to the computer lab and designed to test both conceptual skills as well as the use of state-of-the-art programming tools. At the peak of the COVID-19 pandemic, when exams had to be moved from lab to online, we needed to design an online setup for our practical programming exams preserving the competency-oriented approach and its constructive alignment as well as the validity, reliability and fairness of the exams. The key was to use the same online tools that have been introduced for running lectures and practical classes offering almost the same learning experience as before the pandemic. However, to ensure the validity and fairness of the exams, some kind of online supervision needed to be implemented as technical solutions were found to be either unusable or not working properly in our case. This paper discusses the driving factors, the resulting technical and organizational setup as well as students' feedback and lessons learned for further improvements. Therefore, COVID-19 has not been able to ruin our competency-oriented programming exams.

Keywords: Competency-Oriented Exams, Online Programming Exams, Constructive Alignment, Exams with Third-Party Applications, Online Supervision

### 1 INTRODUCTION

Almost a decade ago, we redesigned our first-year programming courses (GDP1 and GDP2 at our university, internationally often referred to as CS100/101 courses) in a stringent competency-oriented manner. We now strictly focus on the key competencies of (1) understanding a problem statement, (2) modelling the required data and deriving an algorithm to solve it, and (3) implementing and testing the resulting Java program using Eclipse as state-of-the-art Integrated Development Environment (IDE).

In addition to introducing the social learning platform Perusall [1,2] in order to get students better prepared for the lectures, enforce social interaction between peers (which later became quite valuable and important during the pandemic) and leave more of the precious contact time for in-depth discussion of "hard topics" (as identified by Perusall's "Confusion Report") and the use of "Just-in-Time Teaching (JITT)" methods [3] like Peer Instruction or Flipped Classroom, the weekly practical classes were designed so as to apply the newly introduced programming concepts (from data types, control structures, functional abstraction through object-oriented design and programming to GUI development using JavaFX, concurrency/threads etc.) to given programming problems of increasing complexity but at all times focusing on the full-range task from problem understanding to successful implementation. According to Bloom's Revised Taxonomy of Learning Domains [4], the required cognitive task falls into the highest category ("creating") – or level 5 ("synthesis") in Bloom's original taxonomy [5].

Consequently, in order to ensure the 'constructive alignment' of learning and assessment according to Biggs & Tang [6], students are informed from the very beginning of their classes, that it is exactly this competency that is getting tested in the final exams [7]. There are no questions like naming all primitive data types in Java or transforming a for-loop in an equivalent while-loop, but knowledge about data types or loop-statements is nevertheless essential to solve the given programming exercise.

Such competency-oriented exams offer a wide range of advantages over more abstract or more theoretical exam setups. They improve students' learning motivation, as well as their perception of the

# Projektidee: „Challenges“

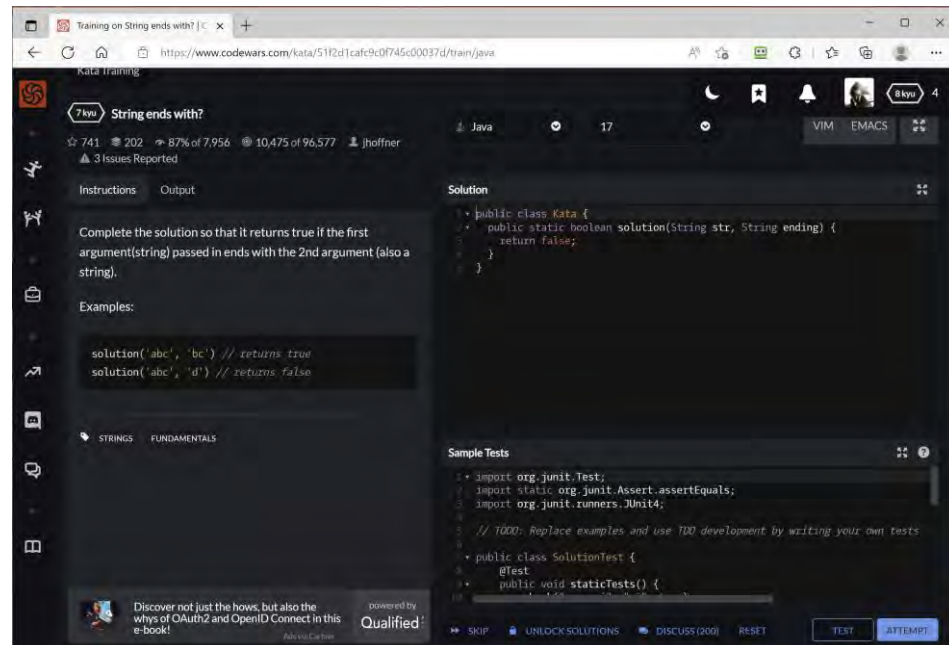
## ■ Grundidee:

Studierende stellen ihren Peers über eine zu schaffende Plattform Programmieraufgaben („Challenges“) – inspiriert durch *codewars.com*.



```
codewars  
This code does not execute properly. Try to figure out why.  
public class Multiply {  
    public static Double multiply(Double a, Double b) {  
        return a * b  
    }  
}
```

SUBMIT



```
Training on String ends with? | C x +  
https://www.codewars.com/kata/5112d1cafe9c0f745c00037d/train/java  
Kata Training  
7kyn String ends with?  
741 202 87% of 7,956 10,475 of 96,577 jhoffner  
3 Issues Reported  
Instructions Output  
Complete the solution so that it returns true if the first argument(string) passed in ends with the 2nd argument (also a string).  
Examples:  
solution('abc', 'bc') // returns true  
solution('abc', 'd') // returns false  
STRINGS FUNDAMENTALS  
Discover not just the hows, but also the whys of OAuth2 and OpenID Connect in this e-book! powered by Qualified!  
SKIP UNLOCK SOLUTIONS DISCUSS(200) RESET TEST ATTEMPT  
Solution  
public class Kata {  
    public static boolean solution(String str, String ending) {  
        return false;  
    }  
}  
Sample Tests  
import org.junit.Test;  
import static org.junit.Assert.assertEquals;  
import org.junit.runners.JUnit4;  
// TODO: Replace examples and use TDD development by writing your own tests  
public class SolutionTest {  
    @Test  
    public void staticTests() {
```

Einsatz in der Lehre:

Das *Einstellen* von „guten“  
(herausfordernd & lösbar)

Challenges wird ebenso

belohnt wie das *Lösen* der von anderen Peers gestellten Challenges.





## ■ Ziele:

Höhere Motivation durch Wettbewerbscharakter (vgl. PrimeGame)

Erstellen „guter“ Challenges erfordert tieferes Verständnis der Thematik.

Das Bearbeiten und Lösen der semesterbegleitenden Challenges dient als formative Prüfung.

Semesterbegleitende Leistung („Continuous Assessment“) reduziert Druck durch finale, semesterabschließende (summative) Modulprüfung.

## ■ Längerfristige Perspektive:

Vollständiger Ersatz der summativen Modulprüfung durch semesterbegleitende und PO-konforme (!) Prüfungselemente, die teilweise durch Studierende selbst gestaltet werden -> „Continuous Peer Assessment“



**Sehr gerne...**



**Manfred Meyer  
Westfälische Hochschule  
FB 6, Bocholt**

**[manfred.meyer@w-hs.de](mailto:manfred.meyer@w-hs.de)**