Elective Course OOP II: Creation, Experiences, Students’ Opinions

Mirjana Ivanović, Dejan Mitrović, Miloš Radovanović

Department of Mathematics and Informatics
Faculty of Sciences
University of Novi Sad, Serbia

Agenda

- Introduction
  - The topics
  - Assignments and grading
  - Students’ opinions
  - Discussions
Introduction

- Object-oriented programming I: existing, mandatory course for all CS students, 3rd semester
  - Covers OO concepts, the Java language, some Java SE libraries

- Object-oriented programming II: elective course, 4th semester

- OOP II is the continuation OOP I, with the focus on advanced Java topics
  - The main goal was to teach the students practical Java skills required by the industry

Agenda

- Introduction
- The topics
  - Assignments and grading
  - Students’ opinions
  - Discussions
The interdependency of topics

- New Java 5 lang. features
- Enumerations
- Generics
- JFC/Swing
- Strings
- Collections
- OO design
- I/O system
- Network programming
- Distributed programming
- Threads
- Java EE

Topic details (1/5)

- New Java 5 language features
- Strings in Java
- Java enumerated types
**Topic details (2/5)**

- **Generic types in Java**

- **Java collections framework**

- **Java threading features**

**Topic details (3/5)**

- **Creating GUI with JFC/Swing**

- **Customizing JFC/Swing-based GUI**
  - Working with panes of top-level containers. Writing custom layout managers. Component development. Java 2D.
Topic details (4/5)

- The Java I/O system

- Java network programming

- Distributed programming with Java

Topic details (5/5)

- Java EE pt. 1

- Java EE pt. 2
  - Service-Oriented Architectures and web services. Java Server Faces.

- Object-oriented design
  - UML. Design patterns.
Agenda

- Introduction
- The topics

Assignments and grading

- Students’ opinions
- Discussions

Assignments and grading

- A student could score the maximum of 100 points, attained from:
  - Practical assignments: 32 points
  - Theoretical tests: 28 points
  - The final exam: 40 points

- In order to qualify for the final exam, a student needed to attain:
  - Min. 16 points (50%) from practical assignments
  - Min. 14 points (50%) from theoretical tests

- The passing grade was 55
Practical assignments

- There were 12 practical assignments in total: 5 individual and 7 group

- Individual assignments covered the following topics:
  - Strings in Java (1 point)
  - Enumerated types (1 point)
  - Generic types in Java (2 points)
  - Java collections framework (2 points)
  - Java threading features (2 points)

- Individual assignments were conducted in a computer laboratory, each within a 90 minute timeframe

Group assignments

- For the remaining 7 practical assignments, students were divided into groups of 3 or 4

- In order to qualify for group assignments, a student had to attain at least 4 points (50%) from individual assignments

- Group assignments covered all the remaining topics, except for the Java I/O system, totaling 24 points

- Each group was given the same task description and had to submit the solution within 1 week
Grading group assignments

- Students had to defend submitted solutions, through individual discussions
- The total number of points given to a student for each assignment was based on several criteria:
  - The overall percentage of the problem that had been solved
  - The amount of work completed by each student
  - The level of understanding the student had for his/hers own code
  - The level of understanding the student had for the code programmed by his/hers coworkers
  - The student’s comprehension of the underlying theory

Grades

- Initially, 53 students enrolled the course
- 28 students (58%) passed individual assignments
- 22 students qualified for the final exam
  - 42% of the initial number
  - 79% of those who had passed individual assignments
- As of June 2011, 13 students passed the final exam:
  - 6: 15%
  - 7: 32%
  - 8: 23%
  - 9: 15%
  - 10: 15%
Agenda

- Introduction
- The topics
- Assignments and grading
- Students’ opinions
- Discussions

Students’ opinions – on topics (1/2)

- How satisfied were you with the course topics?
- How would you describe the difficulty of topics?

![Pie chart showing student satisfaction and difficulty levels.](chart)

- Satisfied: 33%
- Extremely satisfied: 57%
- Extremely unsatisfied: 5%
- Unsatisfied: 5%
- Moderate: 14%
- Very difficult: 14%
- Difficult: 72%
Students’ opinions – on topics (2/2)

- Do you consider the course topics to be valuable for your future profession?

![Pie chart showing 100% response]

---

Students’ opinions – on group assignments

- How would you describe the difficulty of group assignments?
  - Easy: 9%
  - Moderate: 45%
  - Difficult: 41%
  - Extremely difficult: 5%

- Would you rather work on the problems individually, in the computer laboratory?
  - Yes: 27%
  - No: 68%
  - Not sure: 5%
Overall positive impressions

- “I believe that this course, without superlatives and exaggerations, is one of the most useful courses in our entire education.”

- “Only for ambitious programmers, difficult, but very useful. The best course up to this point.” (2nd year student)

- “I think we’ve learned a lot of useful stuff that we will be able to employ in the future. The set of chosen topics is phenomenal, difficult, but phenomenal.”

Overall negative impressions

- The number one negative impression: too many assessments of the theoretical knowledge

- Also, no chances for improving the score

- Poor reactions to the UML and Design patterns topic

- “A lot of new, strange, and difficult topics covered very fast. We were expected to do a lot of stuff on our own, which we were not used to.”

- “Way too much work for an elective course.”
Agenda

- Introduction
- The topics
- Assignments and grading
- Students’ opinions
- Discussions

Discussions on group assignments

- Mixed impressions with group assignments
- Students were (surprisingly) very honest about the amount of work they have (not) completed
- However, in many cases they were not able to divide the work equally
- Irresponsible behavior of some students towards other members of their group
- Overall, grading group assignments correctly was a difficult task
Problems with Java EE

- A large portion of students were simply not able to setup GlassFish with Eclipse on their home computers.

- Students had many small problems while working on the Java EE assignments, but from which they were not able to recover on their own.

- So there were a lot of e-mails exchanged back and forth:
  - A lot of reading of GlassFish-generated exception stack traces.

Future improvements of the course

- Enrich lectures with more practical examples.

- Reduce the impact of the theoretical tests on the final grade:
  - Currently, it is more than 70%.

- Update the course with recent developments of Java language and technology.

- Add new topics?
  - Automated testing appears to be a good candidate.
Thank You for your attention!

Any questions? Suggestions?