Course Report, IOOPM HT16
by: Tobias Wrigstad, with input from Elias Castegren and Stephan Brandauer
January, 2017

To encourage you to save trees, figures are ridiculously small. Use the zoom function or a large screen.

Course Information

Term: HT 2016
Programme: IT and DV
Active Students: ≈ 110
Replies to Survey: 80

Examination Results

They are not known at the time this report is written. Items marked (*) below are still being marked.

- Passed Part 1: 92 students
- Passed Part 2: 82 students
- Passed Part 3: n/a (*)
- Passed C Exam: 93 (*)
- Passed Java Exam: 64 (*)

Strong Sides, According to Students

1. Students have learned a lot.
2. Access to dedicated teaching team which was also adaptive to requests.
3. Several mentions of lectures and the way the course was structured.

Weak Sides, According to Students

1. Not enough lab assistants/too long queue times in the lab.
2. The course is too much work for the credits.
3. Project (specification?) is too hard.

About this Course Report

The course survey is 53 questions long. 80 students completed it. About 40 took the time to write comments. There is lots of additional information in the course survey that is not covered here. Also, this course report is not a report on how to fix things—not enough time has passed to have a handle
on that, plus we usually fix things in the course by working together with the students of last year and the upcoming year, so it would be bad to decide on things too early.

**Teacher’s Comments On Course Organisation**

1. The course organisation ran smoothly this year. A few times there were too few TAs during labs, but better than ever before. This is usually related to people getting sick, which happens in this period every year. Especially with this much face-to-face in the labs. This year we tried to off-load the computer halls with additional spaces open for people with laptops. We’ll try to do that in a less ad hoc way next year.

2. Extra labs in January were scheduled, as usual, in December. Late scheduling of these labs seems to cause some stress but we continue to believe that if they were on the schedule from day 1, we would just have more people asking for extra labs towards the end.

**Teacher’s Comments On Improvements**

Students think the Java part is way easier than the C part of the course. There are probably several reasons for this: students have learned the course’s unusual structure; students have learned to navigate everything that’s new in the C part (git, AUPortal, course web page, terminal, the C syntax); there is a considerable overlap in content (not just while loops, but the whole imperative programming concept).

That said, there is a strong drop-off of students passing the Java exam.

The conclusion we draw is that Java/OOP needs to be given more attention during the course. Especially point out that the lectures cover things that might not always show up in the assignments, or that they won’t use automatically in the assignments if they do not know them. We need to “raise the status of Java”, and possibly reshuffle things to give Java more time. One option might be to do the project in Java, which would sadly negate some positive aspects of the project.

There are many smaller things that can be improved. I list some here:

- "Clarifications/decisions" that get made in Piazza are not properly documented. We should use the pinning feature more and possibly have a page on the course web pages to record these.
- Some goal descriptions can be improved.
- The specification for the project is hard, should be hard but is possibly also too hard. Even though they have been repeatedly told to not optimise prematurely throughout the course, seemingly all groups have fallen into the optimisation trap in the project, just because of how the specification is (purposely) written. Some rethinking might be needed.
The Java exams were not perfect this year. The December one suffered from a common misleading compiler error message. Following suggestions from students unhappy with that exam I made a quite different exam for January that seemed to be even harder. To some extent, this is probably related to not giving enough time to Java.

We removed some requirements for the 2nd C assignment. This was unfair for the students that had already implemented those requirements. Next year, the C assignments should be pruned and possibly some more structure be given.

Assignments (Fig. 4)

Clearly, students feel that the C assignments are not as interesting as the Java ones. C being such a low-level system and the focus on the programming being more library development that library use makes this hard to avoid. That said, it would probably be possible to come up with a more engaging assignment. Given time, this will happen.

Several students mentioned that the assignments were harder than they thought in the sense that there were always more things to fix. Some commented negatively on fixing memory errors in C.

The fact that assignment 2 builds on assignment 1 makes it hard for those that struggled and wrote crap code for assignment 1. One possible remedy for that is to move to a different assignment, but it is also part of the course goals to use code over longer stretches of time. One possible remedy for that might be to develop some library late in the C part that is reused in the project. This will require more thinking.

Lectures (Fig. 5)

Lectures have been moderately well-attended. As has been the trend, there is a significant drop in attendance when the Java part starts. In general students want more live coding, but many also comment that screen casts are better than live coding because of the ability to replay.

10% of all students think that video lectures should replace regular lectures.
Another 10% think that video lectures could be used to replace some lectures. 
49% think that video lectures should be used as a complement to existing lectures.

Some students commented that if one missed the live coding, there was no way to make up for it. Recorded live coding sessions might make up for this.

I would love to have the time to record more material. I don’t feel the lectures are good to record as is, but more screencast-like stuff can be recorded.

**Project (Fig. 6)**

Students seem to feel that the course prepares them moderately well for the project. That is by design and will not change as part of the project is about learning how to learn new stuff, learning that programs don’t exist in a vacuum (domains—stacks, alignments, etc.) and portability. The question we ask on the survey explicitly asks them to not consider softer issues that no one has even remotely tried to teach them.

The project is definitely hard, and may possibly be too hard. The simplest way to deal with this is probably to make the specification easier. The main reason for failure seems to be because students try to implement the optimised version of the specification rather than starting with something that works. 75% of the students believe that the project has deepened their insight into how to develop software as a team and a majority of students want to keep mixing and matching IT and DV.
Code Exam (Fig. 7)

As was mentioned above, students pass the C exam to a higher degree than the Java exam. This can in part be due to the first C exam being very easy (but not too easy), but students in general seem not to have spent enough time with the “interesting” parts of Java that they can manage to pass the code exam.

Students seem to feel that the course prepares them well for the code exams and that the code exams are reasonable. They also generally disagreed with the statement that we should have fewer coding exams in exchange for more elaborate feedback.

Achievement System (Fig. 9)

This year, it is clear that the achievement system has worked well and many students comment on it very positively. Students generally phrase improvements as “merge some achievements” or “more help clustering achievements” (not “replace with something else”). Almost 3/4 of the students report that they have grouped achievements together for demonstration based on how the achievements fit together, which is better than in past years. 64% of all students also think that the setup with constant small checks helped them avoid spurting towards the end. 29% disagree. At the same time 45% of students thinks that their stress-level increased due to the large number of checks.

90% of the students feel that it was always clear to them what grades were reachable to them (66% strongly, 24% less strong). 75% of all students feel that the achievement system has helped them absorb the contents of the course. Being more responsible for driving their own studies (e.g., clustering achievements, planning etc.) is generally perceived as something good.

33% of the students think that the oral examination has greatly influenced their ability to explain, motivate and communicate. 51% agree but to a lesser extent. There is a strong preference for oral immediate feedback over delayed feedback in written form.

40% of the students don’t think that they would have gotten more out of the course by doing fewer oral checks. 26% think that they would have gotten a little more. 18% think they would have gotten a lot more.
Oral Checks

As usual, there are complaints on the queuing time for oral checks. We have improved lots since last year, but the amount of complaints are about the same. The average waiting time was 53 minutes from asking to demonstrate to having the grade entered into the system (so this includes demonstration time). Some students comment that waiting was not a problem because they always had stuff to do. We should try to help the students find a more productive way to spend the inevitable waiting time. We could also try other ways of running checks, like having bookable slots throughout the entire week. This would however likely increase the stress of failing because of the longer waiting time to re-check.

Students have pointed out, both during the course and in the course survey that TAs vary—both in the requirements they place on students, but also their ability to feedback. We continue to believe that this is not a big problem given that each student does an average of 16 demonstrations throughout the course (median 17, number taken from students that have taken 25 or more achievements which is not enough to pass), and have no means to influence what TA they get out of a pool of more than ten. This corresponds to 50% of all students agreeing that variations in TAs isn’t really a problem with 20% disagreeing. A total number of 4 students (5% of responses) think they have not been treated the same as other people they have done demonstrations with. This is sad, but thankfully a low number. In the comment section, a student identifying as female says that she would not get as many questions from examiners as guys she was paired with in the demonstrations. Another student of unknown gender says that it was easier to get a pass if one of the students demonstrating was female.

Other Things

Students are less enthusiastic about the planning meetings this year as they were in the last year (to an extent that was surprising to us then). Many thinks it is a waste of time. We continue to believe that the planning meetings are especially good for catching students that are drifting, and to help lower stress, and stress planning throughout the course. Especially for “weak students”.

One possible future direction for these meetings is to make only very few of
them mandatory (say one at the end of each phase) and offer other meetings that are bookable.

With respect to satisfaction with the way of managing deadlines with both hard and soft deadlines, 60% of all students tick the top two boxes. 71% are in favour of keeping soft/hard deadlines. 19% don’t care and 3% thought it was confusing and are in favour of removing the system.

The use of Git and GitHub on PKD is not a satisfactory introduction for IOOPM. This will be discussed with the PKD teachers to see if those tools can be made “even more mandatory.”

There are comments made that people are dropping other courses. I will investigate by checking other courses’ records and talking to people who can help me check to what extent this is happening. I have had student’s tell me “I look at the older students and they all have lots of math courses they have dropped and that don’t seem to be a problem for them following the education.”

**Gender Imbalance in Grades**

During the course, we have continuously checked the gender balance, and it is clear that female students are statistically more likely to not pass assignments and get lower grades. As I (Tobias) have pondered this, I have connected it to the claim (which I believe it to be true, but I do not have any references to back it up at the time of writing this) that women tend to underestimate their abilities and men to overestimate theirs. That would mean that the course is structurally bad for female students. How to proceed with this is unclear.

One possible way to tackle this problem might be to instruct TAs to help female students identify goals that they seem to have (almost) mastered during demonstrations, and generally spend more time encouraging female students to try for higher grades.

Explaining the imbalance at the code exams is harder and will require some more thought. I will also talk to Joachim Parrow who has tackled similar problems in a programming course on the STS programme.
Final Words

If you have read this document, whoever you are, could you *please* send an email to me (cr@wrigstad.com) and tell me about it! Not only would I like your feedback, but—it is totally unclear to me who actually reads these reports, and thus, it is hard to know, both how to structure them but also how much time to spend on them. If I have zero replies before end of 2017, the next report will be much much shorter than this one.