



Report - DD1337 - 2018-04-24

Respondents: 1
Answer Count: 1
Answer Frequency: 100.00 %

Please note that there is only one respondent to this form: the person that performs the course analysis.

Course analysis carried out by (name, e-mail):

Richard Glassey, glassey@kth.se

COURSE DESIGN

Briefly describe the course design (learning activities, examinations) and any changes that have been implemented since the last course offering.

DD1337 is primarily built around formative assessment - a series of 9 weekly assignments with code submission and oral presentations, which combine to give a final course grade. Students are introduced into the basics of programming, using the Java language, with an objects first philosophy. Beyond learning programming, students are introduced to other key software engineering skills such as version control, issue tracking, testing and design concerns. As this course assumes no prior programming experience, it begins with a low entry barrier, which progressively is raised throughout the course.

The most significant change from the previous course offering was a re-write of the assignment texts with former students/teaching assistants finding ways to reduce the ambiguity and increase the clarity of what was expected. Also, efforts were taken to rebalance and reduce some of the exercises where there was a sudden perceived jump in workload. The presentation of each assignment was improved by linking to general course guidelines and experiences that former students felt needed to be made more clear and findable - essentially a manual for the course. Finally, a series of instructional videos were produced in order to help students get on board with using KTH Github and the Bash shell.

THE STUDENT'S WORKLOAD

Does the students' workload correspond to the expected level (40 hours/1.5 credits)? If there is a significant deviation from the expected, what can be the reason?

The course runs over 12 weeks, and whilst the effort varies between different weeks (students observe the gentle start followed by a ramping up of effort as the course advances), students are expected to devote about 15 hrs per week. Part of this time is mandatory (2hr övn), as well as optional (2hr lecture, 2hr lab), so the additional workload (assignment & self study) should be between 9 and 13 hrs. Looking at the sample of students who completed the survey (67 of 167), the majority of the data clusters around 8hrs to 14hrs. As several students remarked, they are not exactly what to report here, however the data meets the expectation when averaging out across all weeks.

That said, some individuals report devoting significant amounts of time to this course, at the possible expense to other courses. This is a recurring issue and one that we (the teaching team) are always trying to find ways to address, such as the assignment rebalancing effort mentioned above. As has been noted, students enjoy the programming topic, so it is natural that more time will be spent in this module.



THE STUDENTS' RESULTS

How well have the students succeeded on the course? If there are significant differences compared to previous course offerings, what can be the reason?

From the students registered on the course (~183), 167 students passed the course, with a range of grades from A to E. This is in line with previous years, and the pass rate is high. This can be rationalised in three ways: (1) this is an introductory course set at a level of no programming background - most students coming to KTH have some interest in technology and many have programmed before, (2) assignments have completion (komplettering) and the focus is on improving students rather than failing on first attempt, and (3) the grading system allows several labs to be failed and pass the course with less than an A grade. Whilst one interpretation is that this is a generous system, every effort is directed towards making sure that students improve and collect the maximum number of passes they can.

OVERALL IMPRESSION OF THE LEARNING ENVIRONMENT

What is your overall impression of the learning environment in the polar diagrams, for example in terms of the students' experience of meaningfulness, comprehensibility and manageability? If there are significant differences between different groups of students, what can be the reason?

On the aspect of Meaningfulness:

-- Q1-3 & 6 all score higher than 6.0 / 7.

-- Q4 (stimulation) & 5 (belonging) came in lower at 5.8, which seems fair as the exercises are set without too much freedom, and the work is conducted individually. In terms of gender differences, on Q4 females reported 6.6, which was quite higher than for male responses. Why there was this difference is not so clear as there were no additional comments.

On the aspect of Comprehensibility:

-- Q7, 12 & 15 had the lowest scores in the this range of questions (5.7). Q7 & Q12 relate to the intended learning outcomes and how they were assessed. One suggestion here is that the ILOs are now stranded on a page, linked to by KTH Social, which is no longer used, and as such students may not be connecting with this information. One solution could be to bring the ILOs more clearly into the Canvas page, and map the assessments to the ILOs.

On the aspect of Manageability:

Q18 (time to reflect) and Q20 (opportunities to choose) had the lowest scores (both 5.2). Part of the downside of the continuous assessment used in DD1337 is that the pace is relentless and students are only getting a break in the mid-period, which is occupied by math exams.

Furthermore, as mentioned earlier, the assignments are quite proscriptive and as such there is not too much opportunity to choose what to do.

ANALYSIS OF THE LEARNING ENVIRONMENT

Can you identify some stronger or weaker areas of the learning environment in the polar diagram - or in the response to each statement - respectively? Do they have an explanation?

Overall, the results are consistent with previous rounds. On the some questions (opportunities to choose, belonging, and time to reflect) are a reflection of the design, rather than a significant problem. The benefits of continuous assessment come through in open remarks about the course, and students like the regularity and consistency of the module. However, this does limit the opportunities for choice. As the assignments are all programming solutions, there already is quite a lot of freedom in the development of a solution. In previous rounds, optional and more open-ended tasks have been included, but came at the cost of confusion and marking overhead. As students have already communicated that this course occupies a lot of their attention and time, I still feel the current design is fit for purpose.

The current KTH policy of maintaining three systems (static pages, Social and Canvas) places too many demands on students knowing where to visit and find information. At least in terms of the ILOs, the solution is not satisfactory and hopefully this will change.



ANSWERS TO OPEN QUESTIONS

What emerges in the students' answers to the open questions? Is there any good advice to future course participants that you want to pass on?

In general there are lots of positive comments about the overall design of the course. The team that teach the course take pride in making it a good first experience of KTH. We appreciate that it is a lot of effort, and student responses do raise concerns about the levelling up required from the basic weeks to the later weeks. Overall, the students like their TAs, although as some comments suggest, more work needs to be done in ensuring professionalism. Part of the problem here is that information is not being fed back to the course leader who is in a position to make changes.

There were a lot of comments on introducing the students to more tools. The basic tool used is good for beginners, but students are keen to move on to more complex tools. This is a very tricky topic because we want to get the fundamentals in place, before introducing the bells and whistles. A solid developer can survive with a terminal environment. Adding layers of complexity too early is undesirable, and efforts are made to critically analyse the tools so that students do hear the opinions and importance of getting the basics right first. The sister course DD1338 which follows encourages students more proactively to explore other tools. I think this one will rage on for some time :)

PRIORITY COURSE DEVELOPMENT

What aspects of the course should primarily be developed? How could these aspects be developed in the short or long term?

There are some good suggestions about balancing the pace of the course. The first few weeks could have a little more work, so there is not as much as a leap in the later parts. At the same time, the later parts of the course can be revisited to remove some of the more time-consuming elements.

A better link to the ILOs and assignments is needed. These could be reproduced in the course manual that is linked in every assignment in order to bypass the multiple KTH systems.

Finally, in order to make more clear what a passing standard is for an assignment is, for students and TAs, grading criteria will be introduced as a todo list in each assignment that students should check before their final submission.

Course data 2018-04-24

DD1337 - Programming, HT 2017 prginda17

Course facts

Course start:	2017 w.35
Course end:	2017 w.47
Credits:	7,0
Examination:	HEM1 - Assignment, 5.5, Grading scale: A, B, C, D, E, FX, F LAB1 - Laboratory, 1.5, Grading scale: P, F
Grading scale:	A, B, C, D, E, FX, F

Staff

Examiner: Richard Glassey <glassey@kth.se>

Course responsible teacher: Richard Glassey <glassey@kth.se>

Teachers:

Assistants:

- Simon Larsén <slarse@kth.se>
- Adibbin Haider <adibbin@kth.se>
- Sacharias Sjöqvist <sacsjo@kth.se>
- Kwabena Asante-Poku <kwap@kth.se>
- August Bergman <augber@kth.se>
- Per Nyberg <pernyb@kth.se>
- Simon Lindholm <slindhol@kth.se>
- Andreas Wallström <awallst@kth.se>
- Tomas Möre <tmöre@kth.se>
- Marcus Dicander <dicander@kth.se>
- Mustafa Dinler <mdinler@kth.se>
- Sandra Bäckström <sanbac@kth.se>
- Alexander Viklund <viklu@kth.se>
- Kasim Taskin <kasimt@kth.se>

Number of students on the course offering

First-time registered: 168

Total number of registered: 172

Achievements (only first-time registered students)

Pass rate¹ [%] 94.60%

Performance rate² [%] 95.10%

Grade distribution³ [%, number]

- A 83% (132)
- B 6% (9)
- C 6% (10)
- D 2% (3)
- E 3% (5)

- 1 Percentage approved students
- 2 Percentage achieved credits
- 3 Distribution of grades among the approved students