Informatics Course Feedback:
Software Engineering Large Practical (SELP)

The students at the SELP lecture on 25th October 2017 were given a mid-semester course feedback form to complete. This form prompts the students to identify things to keep doing, to start, and to stop, in response to the rubric below.

> *Please use the sections below to give brief feedback on your experience of the course so far. All parts are optional: this is useful information to help improve the course, but you don't need to answer anything you don't want to and all contributions are anonymous. Please be specific and constructive. A single line is enough, but feel free to use the back of the sheet if you need more space.*

Being a practical, SELP has a relatively small number of lectures (about 8). The Piazza forum is used for questions and answers outside of the lectures. The course lecturer has an office hour each week. There are no tutorials or labs for the course.

Fifteen responses were received and are summarised below. Insertions to the student's answers are marked in square brackets [like this]. Similar answers are combined with the vertical bar symbol |.

Stephen Gilmore
25th October 2017

About the course

Keep doing this! It's working well

- *Putting useful code on the slides | showing code examples in the slides | writing sample code on slides | code examples | providing good examples on slides | giving lots of code examples.*
- *Explaining things relevant to the app development, with relevant examples. | Giving clear examples. Explaining all the details. | I like that the lectures are quite detailed.*
- *The app specification is great. The app idea is awesome.*
- *Picking engaging projects for the Large Practicals and keeping specifications generic to unleash individual creativity.*
- *Having a relaxed and welcoming attitude to questions/issues. | Having an extremely caring attitude.*
- *Keep being friendly and lectures are good. Piazza is also good.*
- *Responding accurately and on time in the Piazza forum or in person!*
- *Providing a lot of information. Not leaving students on their own. | Providing a lot of resources and references. | Beneficial links associated with the topics on various things related to the course.*

Thank you for the kind words about the course and the lectures. From my point of view, the course is an absolute pleasure to teach and I was very happy to read these responses which indicate that people are finding it to be a positive experience. Because this survey is conducted in a lecture I'm not sure whether this is entirely representative of *all* of the class because if people are not enjoying a course then they tend to stop coming to the lectures, and so they would not be being sampled here.

Because the SELP is a software development project which uses a real-world development environment (Android Studio) I think that the course's Piazza forum has a more important role to play here than in more theoretical courses. This is because getting some relatively unimportant implementation detail wrong can completely block your progress on the project when your code won't compile (or won't run). It's important to get around these roadblocks as quickly as possible, so I'll be keeping a close eye on the Piazza forum as the course progresses and I'll try to respond to questions as quickly as I can.

Start this! I think it's worth a try

- *Release the lecture topics in advance so as to avoid independent research beforehand.*
- *Provide more constraints on the design of the app; creativity is born from constraints.*
- *Some discussion about how to design using XML and how to use the [Android Studio] screen designer.*
- *Organise the lectures in a way to put first the ones related to the production of the design document and later about the implementation.*
- *Provide examples/advice on how to write a design document.*
- *It was unclear that the same kind of parsing [as] for XML could be used for KML.*
- *I think that it would be cool if the course involved building one or two smaller apps to help us get a grip on how to build Android apps, and after that build [the app for assessment].*
- *Providing example material to show best case approaches (such as an example design document to help understand how to construct it fully).*
- *Labs where we could take our apps and solve problems / get advice.* | *Maybe lab sessions?* | *Labs? Tutorials?*
- *Make classes more interactive e.g. bring in TopHat [electronic voting system].*
- *Live demos.*

I think that there are some fantastic suggestions here, many of which I would never have thought of. Some of these are obviously only relevant for next year's offering of the course but those that apply to this year I can certainly act on. Regarding having lab sessions or tutorials I think that labs seem to be the better option, because of the practical nature of the course. I'll arrange some for later in the semester.

Stop this! I don't find it helpful

- *I have mixed [feelings] about the lectures.*
- *Nothing really, the course is good.*
- *Reiterating the same point three or more times.*

I also have mixed feelings about the lectures, and I don't at all believe that I have the perfect set of lecture topics, but I think that some students could feel quite abandoned if we had no lectures. Someone last year said that only two of the lectures were useful, but unfortunately they didn't say which two! I definitely recognise myself in reiterating the same point three or more times; I'll try to cut back to only two or more times.

About you

What steps can you take to improve your own learning in this course?

- *Use Google, work lots in my own time.*
- *Devote more time to the course. | Spend more time.*
- *Find more time to explore the additional material suggested, such as [using ConstraintLayout] to improve the project.*
- *I am learning to use so many new APIs in this course, given that I have never programmed an app before. Maybe I have to stop trying to understand what happens under the hood and just use the API.*
- *Maybe recap some of my Java skills. | Brush up on my Java.*
- *Make sure to stick to my schedule.*
- *[Think] more about how things work together, [such as] Java [classes and packages] that are used in Android development.*
- *Read documentation and work on the code examples.*
- *Avoiding some problems on DICE; although this is not because of the organisation.*
- *Start the coursework.*
- *Have a more clear plan. Right now it looks like a very large amount of work.*

These are all good strategies for improving your learning, and learning strategically. Specifically, I always try to encourage students to learn Java seriously because it is a skill that you are likely to be able to draw on again in the future, even if you never return to Android development again after this practical. It's correct that the project is a large amount of work and will take a significant amount of time to complete but SELP is a 20 credits course, so this is to be expected.

The answers to the final question on the level of difficulty of the course are summarised in Figure 1. I think that this is a reasonable distribution of answers. I'm glad that no-one (in the respondents surveyed here) is finding the course too easy, because that would be wasting a valuable opportunity to improve your skills.

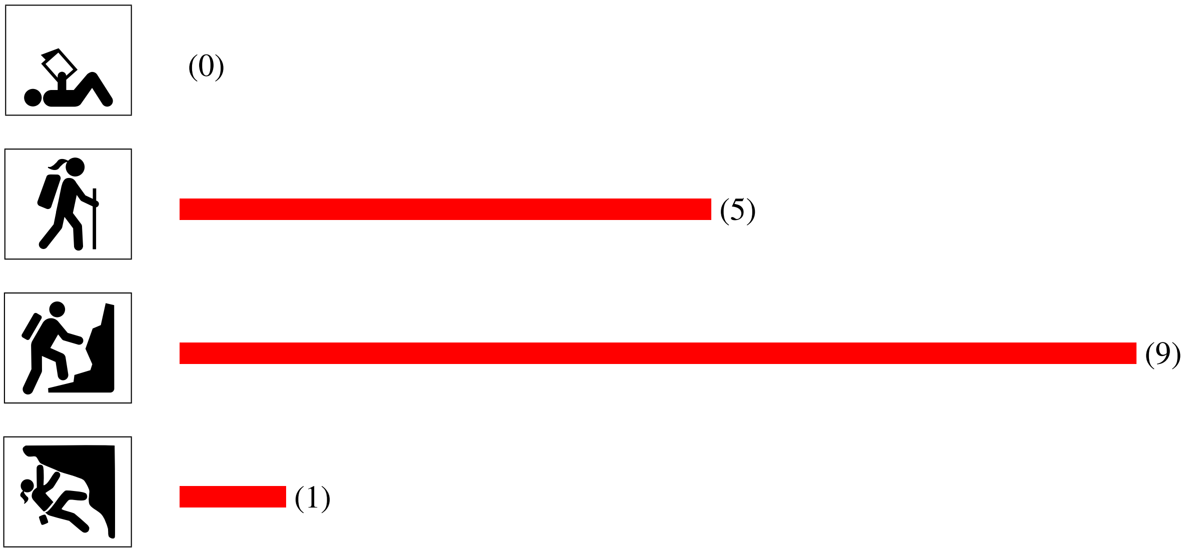

Figure 1: Distribution of the answers to the question "Which of these best expresses the level of challenge you find in this course?"