Informatics Course Feedback: Computer Science Large Practical (CSLP)

The students at the CSLP lecture on 27th 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, CSLP 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.

Twenty-one 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
27th October 2017

About the course

Keep doing this! It's working well

- *I think building an Android app is a great activity to do – I had always viewed it as a scary, difficult thing I would not be capable of but it is actually OK so far.*
- *Lectures are informative and concepts are well explained.*
- *Lectures are helpful and the areas they touch on are probably the most helpful.*
- *Interesting lectures and helpful code samples provided.*
- *Going through one specific topic each lecture.*
- *Android programming slides are very interesting and informative!*
- *Giving code samples in lecture slides is helpful.*
- *Going through code in lectures; good explanation of parts shown in lectures.*
- *Helpful slides; makes the coding easier as the examples and pointers help.*
- *Going over actual examples showing how a feature / activity in Android Studio works.*
- *Good slides with code in detail.* | *Showing code examples.* | *Including code samples to get us started.*
- *Explaining the more in-depth behaviours of modules / features.*

- *Providing links at the end of the slides.*
- *Demos in lectures; examples.*
- *Helping at Piazza. Thanks* :)
- *Everything is working well currently.* | *Everything.*
- *Generally no complaints.*

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.

Start this! I think it's worth a try

- *A Piazza page is really useful in other courses and would be a helpful addition in this course too.* | *[Have a] Piazza forum.* | *[Have a] Piazza forum? Or a place to ask questions.*
- *More hands-on use of Android Studio in lectures.* | *Show examples of writing / running code in Android Studio in lectures.* | *More live demos during lectures.* | *I think using Android Studio to demonstrate the code is a more vivid and impressive way.* | *Writing example code live and executing [it].*
- *Two lectures a week? I just feel like we need more information at the start of the project to get properly started on it.* | *Two lectures a week / lecture + smaller discussion.*
- *More structured goals for the project/learning at the beginning of the course.*
- *More about programming in Kotlin, maybe also about good programming practices.*
- *Provide or point to a few example apps in Kotlin, as documentation still feels a little thin.*
- *Is it possible to provide simple examples for common Android tools in Kotlin such as Broadcast-Receiver, ContentProvider? This would be easy to code.*
- *It would be helpful to include info on requirements for running the Android emulator and alternatives. My computer (Mac, Yosemite) couldn't run it and I wasted days thinking it was me. Genymotion is working well.*
- *[Be] a little more specific on the design document requirements.*
- *An example design document. This would help us as its a little difficult to remember everything we need to do and how to lay out our design document etc.*
- *We really need an example design document.*
- *Lecture topic posted ahead of time.* | *List future lecture topics on course page.*
- *Would like more contact hours for extra guidance.*
- *Some form of support for if people are very stuck.*

- *Give us topics that will be covered, so that we may focus on learning others on our own. Otherwise there is an incentive not to play with Android Studio as we don't want to learn things twice!*

I must admit that I hadn't realised that people didn't know that the course has a Piazza forum. When I read these comments I posted on the `cslp-students` mailing list to advertise the existence of the Piazza forum. I also hadn't realised that people wanted more Android Studio demos in the lectures; I will dedicate an upcoming lecture to an Android Studio demo, with a specific focus on Kotlin programming, Kotlin code quality, and debugging. One problem with on-line demos in lectures is that there is not really a useful artefact left behind afterwards so people who are not at the lecture might miss out significantly. This will not be a problem next year, though, when lectures on the course will be recorded.

The lectures are advertised ahead of time in the sense that the lecture slides are uploaded to the course web site at least 24 hours in advance of the lecture but it might be the case that people wanted a view of the lecture plan at the start of the semester. Again, this is something that can be addressed next year.

As regards the comment about BroadcastReceiver and ContentProvider, an example of BroadcastReceiver was provided in Lecture 4 of the course, and ContentProvider is not relevant for this practical.

Stop this! I don't find it helpful

- *Live demos are probably not worth it.*
- *Sometimes the lectures are very technical and assume prior knowledge, so they are covered very quickly and I get a bit lost.*
- *Since Kotlin is so new it's incredibly difficult to find examples that work!!*
- *SELP and CSLP being the same project. I chose CSLP because I expected it to be more algorithmic and less GUI / user-based. | I just don't think that the idea to have both SELP and CSLP be basically the same thing was a good idea.*
- *Your code frameworks to give a starting point for location services are broken.*
- *Having to use a beta version of Android Studio is a pain.*

There are some interesting comments here. I agree that because Kotlin is a relatively new programming language there are a lot fewer examples of it being used which are available to study but Computer Science should be an up-to-date subject so we should expect to use the latest technology sometimes. Regarding using beta software, I had hoped that the official release of Android Studio 3.0 would be available at the start of the semester, but it just wasn't. However, it is available now so the vast majority of people's development work does not have to use beta-release software. Thanks for the comment about using location services which caused me to look again at my example code and fix that it was not detecting when the current location could not be obtained. I updated the slides from the relevant lecture (Lecture 3) to use the improved code.

About you

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

- *Take the app one step at a time. Don't try to do it all at once. Don't get frustrated / overwhelmed.*
- *I've been reading online tutorials and exploring SDKs. I am finding the lectures helpful and relevant.*
- *Dedicate more time to it.* | *Dedicate more time to working on it; takes a while to get ideas together.*
- *Try things in Android Studio; learning by practice works best.*
- *Read more Kotlin documentation in order to understand how to use the language to make my app better.*
- *Practice using Kotlin more, [do] online tutorials.*
- *Read more related documents.*
- *Stop stressing about workload.*
- *Don't procrastinate.*
- *Find and study more demos.*
- *Slow down and do the basics well before worrying about bonus features.*
- *Nothing to improve, everything is great!*

These are all good strategies for improving your learning, and learning strategically. I definitely recommend trying things out in Android Studio and tackling the app one feature at a time.

The answers to the final question on the level of difficulty of the course are summarised in Figure 1. Not everyone who returned a questionnaire circled one of the images.

I think that this is a fairly reasonable distribution of answers, although I would have liked to have seen more answers in the second category. 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.

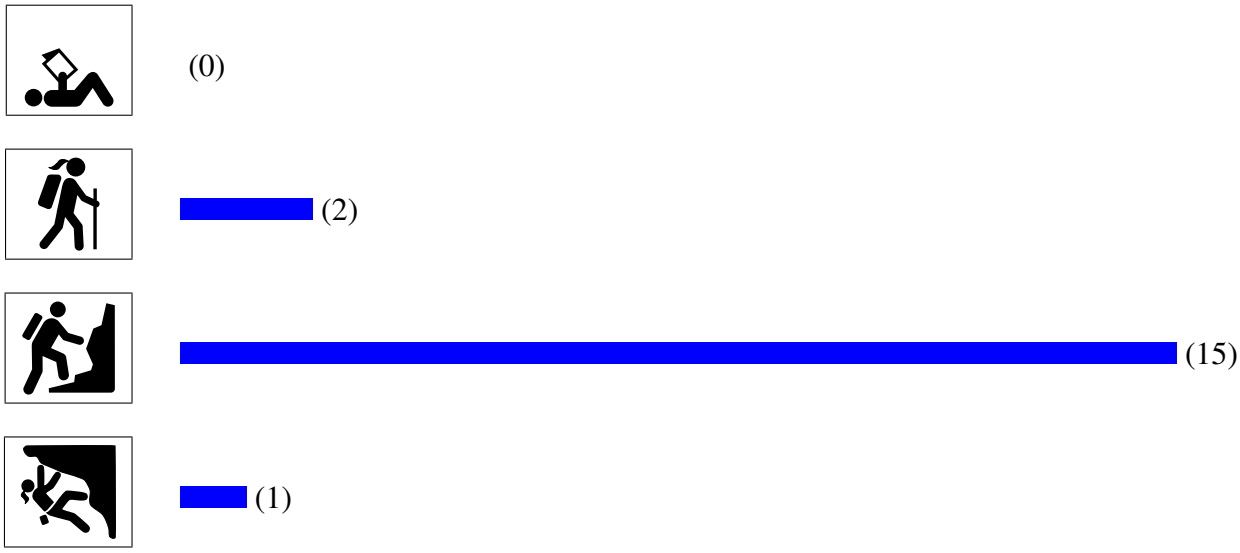

Figure 1: Distribution of the answers to the question “Which of these best expresses the level of challenge you find in this course?”