Assignment 1: Hello Kotlin and Getting to Know You

Identifying Effective Strategies for Learning

Choose a moment in your educational career (it could be an assignment or a full course) where learning went really well. What strategies did you employ that worked particularly well (e.g., working with others, trying work on your own before asking a friend, going to office hours)?

I felt like I had a really great learning experience last semester in ESA + Dynamics + Signals. I reviewed a lot of examples in class during lectures, which helped me work through problem sets much faster. I made sure to always do a “dry run” of a problem set by myself before seeking help or checking answers with others, to make sure I was stretching my brain as much as possible. For help when stuck, I generally compared with friends for an opportunity to both learn and teach (another method of reinforcing learning). Finally, when I had a set of answers I was happy with, I got them checked by a CA during office hours to confirm I hadn’t gotten off track anywhere.

Similar to (1), which sorts of strategies have led to either less effective learning or less enjoyment of the learning experience. Feel free to describe a few examples of what doesn’t work for you.

I have a hard time when flipped classrooms are overstructured; i.e. giving students open work time but interjecting with frequent small examples or lectures. I prefer when open work time is uninterrupted and lecture/example time is in one block; otherwise, I often work faster or slower than a professor expects me to and end up checked out either way. I like having a lot of time in class to work on assignments while spending out-of-class time listening to lectures, watching videos, etc. like a traditional flipped classroom.

As this course is foundational for many aspects of computer science, the problems in this class can be easily solved with modern AI systems (e.g., ChatGPT, Gemini, etc.). One of my foundational assumptions is the process of grappling with a problem helps you internalize the important concepts, gives you more insight into how the tools you are learning can be applied in other contexts, helps you more realistically assess your own abilities, and helps you learn to better communicate your knowledge to others. Particular methods of using AI (e.g., prompting the AI to provide answers to questions and thoughtlessly copying the answers) are unlikely to achieve the learning goals articulated previously. Do you agree with this framing? How are you thinking about AI tools with respect to this course?

I never use AI tools for my classes, not because of any particular moral opposition, but because I simply never started and now have heard a lot of horror stories from people who feel that generative AI has shot their ability to work independently. I expect to continue this habit, for better or for worse.

What strategies will you use in this course to be successful? With respect to AI, what principles or strategies will you use during this course.

My goal will be to do as much of the assignment as I can independently, and time it such that there are one or two blocks of CA hours between that process and the due date. This will give me ample opportunity to check my work with CAs, or just grind an assignment out during office hours when needed.

What do you think of some of the proposed activities for the oral quizzes? Are these activities ones that would give you helpful feedback as to how you are performing with respect to the course material? Would you add or subtract any of the proposed activities?

I love them! No notes! Super awesome! Excited to do them frequently this semester!

How can the teaching team support you?

Recurring CA office hours :)

Hello Kotlin!

What features do you like about Kotlin?

Kotlin feels like a very clean blend of Python and Java to me. I like that the syntax is designed to feel like natural language. I also like the syntax of lambda functions, when statements, and do-while loops. I also like how the language feels designed to be human-readable first and

foremost, rather than being short or space-efficient. I'm curious to know what the utility of using Kotlin over any other language is, besides specifically making Android apps.

Are there things you were expecting to find that you haven't?

I like when statements a lot, they feel like extremely flexible match-cases to me, and I see a lot of application for them.

What questions do you have?

Curious how structuring large Kotlin projects looks! And how complicated is linking separate files and dependencies? Anything like C?

Try using the debugger (see the Getting Set with Kotlin page) for some very basic information on the debugger. Do you have experience using interactive debuggers like this one? Were you able to successfully launch the debugger?

I am generally not a user of interactive debuggers but I really should be, so I look forward to making that a learning goal for myself in this class.