**Theoretical Q & A**

1. In my opinion interfaces are used to define methods in a class so that other unrelated classes could use these methods, and by doing so we can achieve abstraction and inheritance for our OOP project. By achieving abstraction we can hide some unnecessary details from the user, improve our project and keep our code “cleaner”. Main drawback of interfaces is their speed, for instance, abstract classes are faster than interfaces. Interfaces can only declare functionality, unlike abstract classes which permit you to write functionality that can be implemented.
2. We need best practices to improve our quality of life overall. This relates not only to programming but real life events too. As an example you could take a simple thing like greeting your colleagues every morning, it’s not required, but it’s just a thing that is an accepted norm between everyone and it invokes friendliness and communication between each person. For programming we can use keeping our code clean and structured as a best practice. It lets everyone stick to the same principle of writing and understanding written code.
3. I would create tests for each requirement, corner cases, validation errors etc. If I would have the chance I would probably have some other person test out my program because very often the person who develops the program tends to miss out on the errors that they’ve created. It never hurts to have a second pair of eyes on a project.
4. In my opinion the core aspects of being a team member is communication and coordination, no matter if you’re the team leader or a member. If everyone on the team is on the same page of what they’re trying to do, they’ll always achieve it. I think spending time with team members/colleagues outside of work environment is also a key item in having a good team, even going to a bar and spending quality time together provokes better understanding of each other.
5. In my opinion good project structure is as the name implies – structured project. Having classes, models, controllers, etc. labeled and structured in their corresponding folders. Part of the project structure is the source code, it also matters because if everyone wrote code the way that each of them preferred it, the world would be in chaos. If hackers wouldn’t understand the source code, it would be likewise for it’s developers. As I mentioned, project structure matters because if it didn’t, it would make every partaking developers life harder for no reason.