# Aon Pre-Interview Answers: Romeo Senen Albano

### Coding Tasks

**Task # 1: Timing a Method**

Part 1: The task is to be able to have an additional service on the existing class that will measure and log the execution time of each method invocation without changing anything in the existing sample class

SOLUTION: To do that, my approach would be to utilize the Decorator design pattern. This pattern essentially involves creating the same class but extending its methods with an added feature (such as the measure and log service). What does it look like? The 'decorator' class implements the same interface, so the reference type remains abstract. Then, the decorator class will encapsulate an instance of the original and reimplement each method, adding the desired feature. With this, we were able to put an additional feature on the base class without actually modifying the original one. It is safe, flexible, and the original class still exists and is optional

Part 2: Same task but done across multiple classes

SOLUTION: To achieve this, my approach would be to create a master factory for the decorator objects. Each class that we want to have a measure-and-log service will have an existing decorator version of it. Then the factory (statically shared across the application) can easily provide the decorator instances when needed. The idea is that this factory class will act as a utility to the application. This approach is not just flexible; it also gives a sense of optionality to apply the feature only to the specific area of code that needs it.

**Task # 2: Find Arrays**

My approach would be the simplest one. An iterative approach where the smaller array searches each index of the larger array until a match is found.

For this explanation, we’ll call the first parameter arr1 and the second parameter arr2. How it works is that for each element in arr1, starting with the first index (0), we will place the first element of arr2. Then, we will traverse to compare if the arr2 elements match each arr1 element in place. If all matches, then arr2 is a subset, and we return the index where the first element of arr2 is placed as output. Otherwise, proceed to the next index placement. If no more index and no all-match is found, that means arr2 is not a subset of arr1, so we return an invalid index (-1). Moreover, if arr2 is bigger than arr1, then arr2 is definitely not a subset of arr1, so we return an invalid index also.

This approach is not that fast, especially for cases when arr2 is not a subset, but it is simple and easy to understand.

**Task #3: Review a Method**

Below is my feedback:

If I am correct, what you’re trying to do is to have all these conditions in an if block to be true to return a value ‘true’ in the checkStudy method. Below are my thoughts

1. There can be an underlying issue about using == on string comparison. It is not null-safe. Try to use String.equals() method instead. It is by convention the more correct way of comparing string values. You make sure to check for null possibilities.
2. While it works, based on your purpose that everything should be true to return true, we can clean the nested if blocks and convert them into a singular return using the && operator, and omit the return false.
3. If it is assumed that the parameter student will never be null, to improve further, I would advise putting the checkStudy method inside the Student class as a public method instead, so you have direct access to attributes. Also, make “Lee” literal as a final constant in the Student object. This would make the code cleaner and more readable.
4. If not assumed that the parameter student will never be null, then put a null checker if block before calling the object methods for code safety.

## Opinion

**Task #4: Trends**

The technological trend that I have observed has happened in recent years and is a hot topic to this date is the rise of AI tools, automation, and other machine intelligence technologies. Since the launch of Large Language Models like ChatGPT, data generation and processing across industries have never been the same. Many companies are now utilizing these AI tools to create their own co-pilots. Many highly rated products now have AI-assistant features. And I see this trend moving forward, affecting corporate applications towards a new shape of the software development process.

In my current company, software engineers like me are being oriented and mentored to be able to utilize these AI tools in coding. I had the opportunity to be able to integrate Cline (an LLM tool) in my IDE and be able to use it to generate Unit Tests and improve the performance of my code. I believe that the best way to keep yourself on the latest trends is to try and play with the tools yourself. I prepare myself by always keeping an open mind and accepting that technology is never gonna go backwards.

**Task #5: Describe Software that Delivers User Value**

In every project that I have worked on, there’s this one simple product that even the best engineers I know would rather spend the night creating a big chunk of code rather than make this. However, being able to communicate with user clients and user testers, I have learned that this is a highly valued product for them, as this would be the basis to know if the application is really doing what it is tasked to do. It’s called the User Manual.

When I transitioned to Lead Engineer and was able to have visibility on the user verification testing after release, I realized it’s not so bad to being tasked on analyzing the system requirements and converting it into a user-friendly handbook. Being able to create accurate, user-friendly manuals helps a lot for the end users to be able to grasp the application quickly and therefore speeds up their process and contribution to the business. A lousy manual could lead to confusion thus degrading the user experience and can lead to bug reports. User manuals describe the integrity of the project and its processes. The key aspect of creating high value for the user is convenience, if their lives have improved after the project. And any software project wouldn't really achieve that without proper documentation

**Task #6: Your Values**

The core principle that I value the most as a software engineering professional working in a team environment is growth. In an industry that evolves as quickly as software engineering, I realized that the learning opportunity here is infinite. You can learn not just from books or online videos but also from the experience in the work environment and from the knowledge of the team you are collaborating with. It is important to me because I am always in pursuit of new knowledge and experience. Growth is what I am always looking for in every space I stay in.