# Educational Initiatives Interview Experience

| | |
|---|---|
| **Name** | **Karthikeyan S** |
| **Role** | **Software Development Engineer** |
| **Offer** | **Full Time** |
| **Contact** | **karthikeyansevugamurthy@gmail.com** |

**Round 1 – CODING EXERCISE**

**Date:23/07/2024**

**Deadline: 30/07/2024 (7 days)**

Company gave some coding question for shortlisted students based on code implementation and uniqueness of code.

**Exercise 1: Coding on Design Patterns**

Come up creatively with six different use cases to demonstrate your understanding of the following software design patterns by coding the same.

1. Two use cases to demonstrate two behavioural design patterns.
2. Two use cases to demonstrate two creational design patterns.
3. Two use cases to demonstrate two structural design patterns.

I selected two use cases for each of the three categories:

• **Behavioural Design Patterns:** Commands and Interpreter Pattern.

• **Creational Design Patterns:** Singleton and Factory Methods Pattern

• **Structural Design Patterns:** Composite and Flyweight design pattern.

Refer to this GitHub link where I've already pushed the use cases to my repository.

**Exercise 2: Problem Statements for Mini-projects**

I chose problem statement: **Mars Rover Programming Exercise**

**Problem Statement:**

Create a simulation for a Mars Rover that can navigate a grid-based terrain. Your Rover should be able to move forward, turn left, and turn right. You'll need to make

sure that it avoids obstacles and stays within the boundaries of the grid. Remember to use pure Object-Oriented Programming, design patterns, and avoid using if-else conditional constructs.

**Features**

- *Grid-Based Navigation*: The Rover navigates a grid defined by user inputs.
- *Rover Commands*: Supports commands to move forward, turn left, and turn right.
- *Obstacle Detection*: The Rover detects obstacles and records their positions.
- *Status Report*: The Rover provides status reports on its current position and direction.
- *Design Patterns*: Utilizes Command Pattern, Composite Pattern, and other OOP principles for a clean and maintainable codebase.

**Design Patterns**

- *Command Pattern*: Encapsulates 'M', 'L', 'R' commands as objects.
- *Composite Pattern*: Represents the grid and obstacles.
- *Singleton Pattern*: Manages the state of the grid.
- *OOP Principles*: Applies encapsulation, polymorphism, and inheritance effectively.

Refer this link for coding exercises question.

Refer to my GitHub for exercises 1 and 2 link.

## Date:01/08/2024

I've been selected for the next round of the **Technical Interview**.

**Round 2 – TECHNICAL INTERVIEW (OFFLINE)**

Duration: 30-40 minutes.

It was a one-on-one panel interview that began with a self-introduction.

The interviewer quickly moved to a coding question (exercises-2) that had already been provided. He assessed my technical and coding knowledge by asking me to explain the code I had previously uploaded to GitHub. Satisfied with my explanation,

he provided some test cases to evaluate my code, which worked perfectly as he expected. Additionally, he gave me a diagram and more test cases to further test my understanding of the coding question.

He asked about design patterns, specifically Singleton and Flyweight patterns, and I explained them using real-time examples. He was impressed with my explanations of both patterns and the code exercises.

He then reviewed my resume and asked whether I preferred DBMS or DSA, as both were listed in my areas of interest. I said I was comfortable with both, but he wanted me to choose one, so I selected DBMS. He then began asking questions on that topic.

**DBMS:**

I discussed the types of SQL commands and listed the commands for each type, which impressed him. I also explained the seven types of joins in SQL, and he was impressed with that as well. When asked to explain a self-join with a real-time example, I provided a detailed explanation. Additionally, I described transactions and ACID properties with real-world examples, which he found impressive.

Other candidates had additional rounds where they were required to submit an application or optimal code to GitHub or an ER diagram, but I was advanced directly to the HR round.

**Date:01/08/2024**

**Round 3 – HR & TECH INTERVIEW (OFFLINE)**

Duration: 30-40 minutes.

The general interview included three interviewers. The HR representative asked me to introduce myself. They then asked about the Mars Rover Programming Exercise and identified more edge cases than the Technical HR.

Next, he asked questions about OS and DSA. In OS, he inquired about preemption scheduling and how to kill a process. I answered based on my knowledge, and he was satisfied. We then discussed hashing, hash tables, and collision handling, which I explained with examples, and he was pleased with my responses. Finally, he asked general HR questions about my family background, negative traits, and how I plan to overcome them. I received positive feedback overall, though they advised me to reduce over-talking in HR responses.

***Stay motivated and keep believing in yourself—your passion and dedication will guide you to success and to the right opportunity.***

**Note:**

1. Make sure your projects are strong because they show what you know and what you can do.
2. Be strong in basics (DSA, OOPs, SQL, DBMS) {reference links}.
3. Have a solid understanding of design patterns.

**Best of luck with your placements!**