**Ques : Share your perspective on the judicious use of exceptions in Java applications. Describe a scenario where a well-structured exception strategy enhanced code robustness and maintainability.**

Ans : I feel exceptions should be implemented only where the actual requirement is there. Checked exceptions can be used for recoverable errors, and unchecked exceptions for unexpected errors. We need to identify the specific requirements and implement them accordingly. I feel we also need to provide clear and specific messages for exceptions.

**Ques: Explain your approach to identifying code smells or areas for improvement in Java codebases. Provide an example of a refactor you performed that led to better maintainability or readability.**

Ans: I have always followed the SOLID principles for maintaining good code. We have always done regular code review,peer-to-peer review, and used tool like Sonar Lint. We write comments only when its required otherwise code should always be self explanatory.

**Ques : When working with dependencies in Java, what factors guide your choice between different dependency management tools or frameworks? Share an experience where efficient dependency management positively influenced a project.**

Ans : When working with dependencies in Java, I always research first the requirements of that dependency for the project. I prefer those that are compatible with the project, easy to use, popular, and have community support with regular updates.And i also see what other people are saying about the dependency and choose the right one.