1. **What is SDLC? Explain its phases.**

   🔹 ***Answer:***

   SDLC (Software Development Life Cycle) is a structured process for developing software. It ensures software is delivered with high quality, within budget, and on time. Phases:

- Requirement Analysis → gather client requirements
- Design → HLD & LLD, architecture design
- Development → coding in Java (Spring Boot, etc.)
- Testing → unit/integration testing, bug fixing
- Deployment → production release
- Maintenance → enhancements, bug fixes post-release

---

2. **Which SDLC model have you worked in (Waterfall, Agile, Spiral, etc.)?**

   🔹 ***Answer (Infosys often expects Agile):***

   I have worked mainly in the Agile methodology, where we follow Scrum. We divide the project into sprints, usually of 2 weeks. Each sprint includes requirement discussion, coding, unit testing, and review. Daily stand-ups help track progress. This allowed us to deliver working modules frequently and adapt to client feedback quickly.

---

3. **Difference between Waterfall and Agile models?**

   🔹 ***Answer:***

- **Waterfall: Linear, sequential process. Each phase (requirement → design → development → testing → deployment) happens only once. Changes are costly.**
- **Agile: Iterative and incremental. Development happens in sprints with continuous feedback, testing, and integration. More flexible and widely used today.**

Java + SDLC Practical Questions

4. **During the Development phase of SDLC, how do you ensure code quality in Java?**

   🔹 ***Answer:***

- Follow coding standards and best practices (naming conventions, SOLID principles).
- Use Spring Boot for clean API design.
- Write unit tests (JUnit, Mockito).
- Use SonarQube/Checkstyle for static code analysis.
- Peer code reviews.

---

5. Which tools did you use for version control, CI/CD in your SDLC?
   顎 Answer:

- Version Control → Git, GitHub/GitLab
- Build → Maven/Gradle
- CI/CD → Jenkins pipelines for build, test, and deployment
- Testing → JUnit, Postman (for APIs)

---

6. **How are requirements captured and translated into code in your project?**
   顎 *Answer:*

- Requirements were collected in JIRA (user stories).
- We created design documents (HLD/LLD).
- Broke tasks into smaller modules.
- Implemented using Java + Spring Boot REST APIs.
- Tested using JUnit and integrated with SQL database.

Testing & Deployment Questions

7. **How do you perform testing in SDLC for a Java project?**
   顎 *Answer:*

- **Unit Testing: JUnit + Mockito for service layer testing.**
- **Integration Testing: Test APIs with Postman/Rest Assured.**
- **System Testing: End-to-end testing with DB + frontend.**
- **Regression Testing after every deployment.**

---

8. **How do you handle production issues (Maintenance phase of SDLC)?**
   顎 *Answer:*

- Reproduce issue in lower environment.
- Debug logs (log4j/slf4j) for root cause.
- Fix in code and test with regression.
- Deploy patch through CI/CD pipeline.
- Update documentation and communicate with stakeholders.

Advanced / Situational Questions

9. Explain a real-time scenario where Agile SDLC helped in your Java project.

顎 *Answer:*

In my last project, a client frequently changed requirements for a smart ship monitoring system. Since we followed Agile, we adjusted the backlog in every sprint. I developed Java REST APIs in Spring Boot and could deliver modules quickly. Continuous client feedback avoided rework and reduced release risk.

---

10. Where does Database (SQL) fit in SDLC phases?

顎 *Answer:*

- **Design Phase → Database schema design (ER diagrams, normalization).**
- **Development Phase → Writing SQL queries, stored procedures, integrating with Java APIs.**
- **Testing Phase → Database unit testing, checking query performance.**
- **Maintenance Phase → Schema updates, performance tuning.**