☐ I. Core Technical Skills

☐ Java Language (Advanced)

- Deep knowledge of Java 8–21 features

- Streams, Lambdas, Functional Interfaces

- Generics, Enums, Annotations

- Multithreading, Concurrency (Executors, ForkJoinPool)

- Memory management and Garbage Collection

- Performance tuning (JVM parameters, profiling)

☐ OOP & Design

- Design patterns (GoF)

- SOLID principles

- Clean Code and refactoring practices

- Domain-Driven Design (DDD)

⬚ Advanced Java APIs

- Collections and Streams API

- Java NIO, File I/O

- Reflection, Annotations

- Serialization

- Java Module System (JPMS)

⬛ II. Frameworks and Libraries

⬛ Backend Development

- Spring Framework:

  - Spring Core (DI, Beans)

  - Spring Boot (Autoconfiguration, starters)

  - Spring MVC (REST APIs)

- Spring Data JPA

- Spring Security (JWT, OAuth2)

- Spring AOP

- Spring Cloud (Config, Discovery, Circuit Breaker)

- Hibernate (JPA implementation)

- MapStruct or ModelMapper

- QueryDSL or JOOQ (advanced query construction)

⬚ Database & Persistence

- SQL (Joins, Indexes, Triggers, Procedures)

- NoSQL (MongoDB, Redis, Cassandra)

- Database schema design and optimization

- Flyway/Liquibase for DB migrations

🔗 Messaging & Integration

- Apache Kafka / RabbitMQ

- RESTful API and OpenAPI/Swagger

- SOAP (JAX-WS, JAXB if relevant)

- gRPC (optional)

⬚ III. Architecture & DevOps

⬚ Software Architecture

- Layered architecture, Hexagonal/Onion

- Microservices vs Monoliths

- Event-driven systems

- CQRS and Event Sourcing

- API Gateways (like Zuul, Spring Cloud Gateway)

🧭 DevOps & Deployment

- Docker

- Kubernetes (basic to intermediate)

- Jenkins / GitHub Actions / GitLab CI

- Helm Charts (optional)

- Monitoring with Prometheus/Grafana

- ELK or EFK stack for logging

☁ Cloud Platforms

- AWS (EC2, S3, RDS, Lambda, ECS, CloudWatch)

- OR GCP / Azure equivalents

- Infrastructure as Code (Terraform, CloudFormation)

🧪 IV. Testing

- JUnit 5, Mockito, AssertJ

- Integration testing (Testcontainers, WireMock)

- Contract testing (Pact)

-

Performance testing (JMeter, Gatling)

- Load testing and profiling tools

⬚ V. Tools & Ecosystem

- Git, GitHub/GitLab/Bitbucket

- Maven or Gradle

- IntelliJ IDEA (preferred)

- SonarQube or Code Quality Tools

- Postman / Insomnia

- Lombok, SLF4J / Logback

- IDE profiling tools and JVisualVM

⬚ VI. Project & Domain Experience

- Large-scale enterprise apps

- Handling legacy code and refactoring

- CI/CD pipelines

- Handling system failures, retries, resilience

-

Monitoring and alerting

- Real-time data processing (Kafka streams, Flink, etc.)

⬚ VII. Soft Skills

- Mentoring juniors

- Code reviews

- Writing technical documentation

- Agile/Scrum participation

- Decision-making and architecture discussions

- Communication with stakeholders

- Time management

⬚ VIII. Certifications (Optional but helpful)

- Oracle Certified Professional: Java SE 11/17 Developer

- Spring Professional Certification

- AWS Certified Developer / Solutions Architect

- Kubernetes Certified Developer (CKAD)

🎓 IX. Portfolio & Interview Preparation

📁 Projects

- At least one large backend system (ideally open-source or deployed)

- GitHub profile with contributions

- Projects demonstrating microservices, DDD, and Spring Boot

🧩 System Design Knowledge

- Scalability principles

- Load balancing, caching, database sharding

- CAP theorem, eventual consistency

🎯 Interview Prep

- Leetcode/Codeforces for coding

- System design mock interviews

- Behavioral questions (STAR method)a