1. Core Java Concepts (Advanced Level)

- **OOP Principles**: Focus on inheritance, polymorphism, encapsulation, and abstraction.
- **Collections Framework**: Understand various collections like List, Set, Map, and their implementation classes. Be able to discuss when to use which.
- **Concurrency and Multithreading**: Know how to manage threads, thread lifecycle, synchronization, locks, and Executor Service.
- **Exception Handling**: Master checked vs. unchecked exceptions, creating custom exceptions, and best practices.
- **Java 8+ Features**: Focus on Streams API, Lambda expressions, Functional Interfaces, and Optional.
- **Java Memory Management**: Heap, Stack, Garbage Collection, memory leaks.
- **String Manipulation**: String operations, StringBuilder, String Buffer.

3. Collections Framework

- **List, Set, Map, Queue**: Understand the differences, use cases, and underlying implementations (ArrayList, LinkedList, HashSet, TreeSet, HashMap, LinkedHashMap, PriorityQueue).
- **Iterators**: Internal vs. external iteration, forEach method.
- **Concurrency in Collections**: ConcurrentHashMap, CopyOnWriteArrayList, blocking queues.

2. Data Structures & Algorithms

- **Sorting & Searching Algorithms**: Merge sort, quicksort, binary search, etc.
- **Data Structures**: Arrays, Linked Lists, Stacks, Queues, Trees, Graphs, and HashMaps.
- **Complexity Analysis**: Learn Big O notation, time complexity, and space complexity for common operations.
- Practice on platforms like **LeetCode**, **HackerRank**, or **Codeforces** for problem-solving skills.

3. Design Patterns & Best Practices

- Learn popular **design patterns** such as Singleton, Factory, Observer, Builder, and Strategy.
- Be familiar with **SOLID Principles** and understand **clean code practices**.

4. System Design (Basic)

- Understand the fundamentals of designing scalable systems.
- Learn about databases, sharding, caching (Redis/Memcached), load balancing, and microservices architecture.
- Prepare to design small to medium-sized systems with diagrams (UML, flowcharts).

5. Databases & SQL

- Master **SQL**: Writing complex queries, joins, subqueries, and indexing.
- Learn about **transactions**, isolation levels, and ACID properties.
- Explore **NoSQL** databases (e.g., MongoDB, Cassandra) for modern applications

3. System Design with AWS

- **AWS EC2**: Learn about instances, load balancing, and autoscaling.
- **S3 and RDS**: Understand storage and database services for handling large datasets.
- **Lambda**: Learn how to deploy serverless applications in AWS.
- **API Gateway**: Understand how to manage and monitor APIs using API Gateway in AWS.

4. Cassandra

- Learn the basics of **Cassandra** architecture: nodes, clusters, replication, and partitioning.
- Understand **CQL** (Cassandra Query Language) and how to design queries for distributed databases.
- Learn about **Cassandra's NoSQL model**: how it handles large data volumes and distributed queries efficiently.

5. Amazon Redshift

- Understand **Amazon Redshift** as a cloud data warehouse service.

6. Jumphost (Bastion Host)

- Understand the role of **Jumphost** (Bastion host) in securing access to internal servers.
- Learn how to set up **SSH tunneling** to access private resources in the cloud environment.
- Understand **best practices** for managing secure connections between external resources and internal systems using a Jumphost.