JAVA

* JVM Memory Model
* OOPs Concepts
* Java 8 Concepts
  + Stream API
  + Functional Interface
  + Functional Programming
  + Lambda Expressions
  + Default Methods
  + Map Implementation Changes
  + Optional
* Collections
  + Map
  + TreeMap
* Multi-Threading
* String
* Exceptions
* Exceptions for Collection Frameworks

Unit Testing and Junit 5

* Unit Test Coverage
* Test Pyramid
* Unit Test Principles and Structures
* Common Annotations
* Conditional Annotations
* Assumptions
* Mocking – Mockito or Power Mock
* Spying
* Stubbing

REST API Development

* Annotations
* Validations
* Security
* REST Maturity Model
* OPEN API Standards

Spring Framework

* Configuration (Annotation, properties, XML)
* IOC/DI In-depth
* Bean Scopes
* Annotations
* Contexts and Context Configuration
* Important Modules
  + Spring Data JPA
  + WEB MVC
  + Test

DESIGN THINKING

Learn or refresh your knowledge of Object-oriented design principles.

Try to apply Object-oriented design principles. The solution should also follow principles like loose coupling, and high cohesion.

Learn Component Cohesion Principles:

1. REP – The Reuse/Release Equivalence Principle
2. CCP – The Common Closure Principle
3. CRP – The Common Reuse Principle

Learn Component Coupling Principles:

1. The Acyclic Dependencies Principle
2. The Stable Dependencies Principle
3. The Stable Abstraction Principle

Learn Below Principles

1. DRY (Do not repeat yourself)
2. KISS (Keep it simple, stupid)
3. YAGNI (You aren’t gonna need it)
4. Law of Demeter – Friend of a Friend
5. Composition over Inheritance
6. Fail-Fast

Separation of Concerns