API Design

1. Introduction to RESTful APIs
    a. What is API?
    b. WWW
    c. JSON & YAML
    d. RESTful Architecture
2. Effective API design general principles
3. API design best practices
4. Introduction to SwaggerIntroduction to Swagger
    a. Datatypes
    b. Schemas
    c. Paths
    d. Components
    e. Error Codes
    f. Security
    g. Request & Response
    h. Tags
5. Hands-on session on writing Demo Facebook APIs

Docker & Kubernetes

1. Introduction to Docker
    a. Evolution of Containerization
    b. VM vs Containers
    c. Why Docker? Why now?
    d. What is docker and Who is it for?
    e. Docker - Advantages & Caveats
    f. Docker Ecosystem & Architecture
2. Installation & Setup
    a. Docker Editions
    b. Docker Versions and updates
    c. Docker for Mac
    d. Docker for windows 10 pro
    e. Docker for windows 8.1
    f. Install Docker on Linux
3. Docker Containers
    a. Checking Docker install and configuration.
    b. List running container.
    c. Overriding default command
    d. Container lifecycle
    e. Restarting stopped containers.
    f. Removing stopped containers.
    g. Remove all containers.
    h. Remove running containers.
    i. Stopping Containers
4. Docker Image
    a. What are Docker images
    b. Exploring Docker Hub
    c. Concept of Docker image layers
    d. Image tagging and pushing to Docker hub.
    e. Building Images - Docker file Basics
    f. Extending Docker Official Image

    g. Building Images - Create custom Docker Image

5. Docker Volumes
    a. Container Lifetime and persistent Data
    b. Persistent data with volumes
    c. Persistent Data with bind mounts
6. Docker Networking
    a. Network Drivers in Docker
    b. Docker Networks-CLI Management
    c. Docker Networks-DNS
7. Docker Compose
    a. Docker Compose Overview
    b. Docker Compose with YAML
    c. Docker Compose CLI
    d. Deploying Web App with Compose

Advance Java

1. Inner Classes
    a. Overview and Motivation
    b. Stronger Encapsulation, Rules, and Caveats
    c. Defining and Using Inner Classes
    d. Member-Level, Method-Local, Anonymous Classes
    e. Static Nested Classes
    f. Nested Classes, Nested Interfaces, Nested Enums

2. Java API Techniques
    a. The Console class.
    b. The String Builder class
    c. Formatting techniques
    d. Regular expressions
3. File Handling
    a. Working with files
    b. Text files
    c. Binary files
    d. Serialization
    e. XML files
    f. Java properties files
4. Localization and Resource Bundles
    a. Locales
    b. Resource bundles
    c. Locale-specific formatting and parsing

5. Multithreading Techniques
    a. Java synchronization language features
    b. Designing thread-safe classes
    c. Recommendations for synchronizing resource access

    d. Using concurrent collections
    e. Using synchronizers and locks
    f. Thread pooling techniques
    g. Using the executor framework
    h. Using pooling effectively

6. Logging Overview
    a. Popular Logging Frameworks (Log4J)
    b. Writing Log Messages
    c. Creating Loggers and Writing Log Messages
    d. Log Levels & best practices

7. Garbage Collection
    a. Essential concepts
    b. Understanding object lifetimes
    c. Generational collectors
    d. Heap organization
    e. Garbage collection options
    f. Garbage collection monitoring and tuning (memory analyzer tool)

8. Lambda Expressions & Functional Interfaces
    a. Overview
    b. Functional Interfaces and Lambdas
    c. Target Context
    d. Using Lambda Expressions
    e. Syntax, Lambda Compatibility
    f. Variable Capture
    g. Type Inference

9. Method References
    a. Three Types of Method References
    b. Refactoring Lambdas into Method References

10. Stream API
    a. Overview
    b. Streams vs. Collections
    c. Anatomy of a Stream
    d. Intermediate Operations and Stream Pipeline
    e. Java 8 Functional Interfaces: Predicate, Comparator, Function, Consumer, Supplier

11. Stream Processing
    a. Filtering, Sorting, Mapping
    b. Terminal Operations
12. Collectors
    a. Partitioning and Grouping
    b. Reducing and Summarizing
    c. Downstream Reductions
    d. Optional Class
    e. Optional. Empty
    f. Optional. Of

13. Database Access with JDBC and JPA
    a. JDBC Overview
    b. JDBC Architecture and API
    c. Using Driver Manager, Connection, Statement, and Result Set
    d. Implementing CRUD operations with MySQL Database
    e. JPA Overview
    f. JPA Architecture and Programming View
    g. Entity Classes and Annotations
    h. Mapping an Entity Class
    i. Entity Manager Factory and Entity Manager
    j. Working with JPA

14. Advanced JDBC Techniques
    a. Data Sources
    b. Metadata
    c. JDBC escape syntax.
    d. Transaction management
    e. Additional techniques
15. TDD and Unit Testing
    a. Principles of unit testing
    b. Using JUnit effectively
    c. Dependency injection and mocking
16. Using Junit
    a. The JUnit framework
    b. How to define a test in JUnit?
    c. Example JUnit test
    d. JUnit naming conventions
    e. Assertions
    f. Creating and Using Text Fixtures
    g. Test Run Lifecycle: @BeforeEach and @AfterEach, @BeforeAll and @AfterAll
    h. JUnit test suites
17. Basic Junit Code Constructs
    a. Available JUnit annotations
    b. Assert statements.
    c. Test execution order
    d. Disabling tests
18. Additional Testing Needs
    a. Testing for Exceptions
    b. Setting Timeouts
    c. Assertion Groups
    d. Characteristics of Good Tests
    e. Writing Testable Code

19. Testing with Mocks – MOCKITO
    a. Overview
    b. Mock Objects as Collaborators
    c. Mockito Overview
    d. Mockito
    e. Creating and Using Mocks
    f. Basic Steps in Mocking
    g. The Mockito Class
    h. Mock Creation with @Mock
    i. JUnit 5 Mockito Extension
    j. Stubbing
    k. Additional Capabilities
    l. Argument Matchers
    m. Partial Mocking with Spies
    n. Mocking the Unlockable
    o. Dependency Injection of Mocks

Design Patterns

1) Design Patterns
    a) History
    b) Benefits
    c) Catalog
    d) Composite
    e) Chain of Responsibilities
2) GOF Behavioural Patterns
    a) Strategy
    b) Command
    c) Observer
    d) Template Method
    e) Iterator
3) GOF Creational Patterns
    a) Factory
    b) Abstract Factory
    c) Builder
    d) Singleton
    e) Prototype
4) GOF Structural Patterns
    a) Façade
    b) Proxy
    c) Composite
    d) Decorator
    e) Adapter
    f) Flyweight