# Core Java

## Java Varargs

## String Constant Pool

## OOP

## Java I/O

## Java Enum

## Java Annotations

## Java Regex

## Java Multithreading

## Java Serialization

## Java autoboxing and unboxing

## Collections Framework

## Wrapper Class in Java

## Java Exception Handling

## Generics

## Functional programming

## Stream

## Multi-Threading

## JAVA Records

# Spring Core Frameworks

* 1. Dispatcher Servlet
  2. The request processing flow of Spring
  3. IOC Container
  4. Dependency Injection
  5. Spring Bean and its life cycle,
  6. Bean Scop, Component, Lazy init, bean config
  7. AOP
  8. Spring Security
  9. DI achievement without Spring
  10. Transaction & Rollback
  11. API, REST, SOAP, RPC, GRPC
  12. Testing, Unit TEST
  13. Internationalization
  14. Spring Boot auto-config
  15. Spring Scheduling
  16. Circular Dependency and how to remove it

# ORM

* 1. Data JPA & Spring Data JDBC
  2. Hibernate Session management

# Database

# Server

# Micro Service Architecture

# Message Queue

# Caching

# Networking

# Containerization

# System Design

# Problem-Solving



**Additional Topics**

1. Java Varargs
2. Java I/O
3. Java Enum
4. Java Annotations
5. Java Regex
6. Java Multithreading
7. Java Serialization
8. Java autoboxing and unboxing
9. Collections Framework
10. Wrapper Class in Java
11. Java Exception Handling

**Java Knowledge Recap:**

## Sprint 1 :

- Why choose Java

- Compiled vs Interpreted Language

- Java Compiler (javac), How it works

- What is classloader

- What is bytecode

- How is Java Platform Independent? What is JVM

- JIT Compiler

- Structural vs OOP

- Writing a Simple Java Class

## Sprint 2:

- Static vs NonStatic

- Java Primitive vs Non-Primitive.

- Why is Java not fully OOP?

- Java Object Creational Analysis in JVM.

- 4 Pillars of OOP (Encapsulation, Inheritance, Abstraction, Polymorphism) in details with use case

- Abstraction vs Polymorphism

- Why String is Immutable in Java, Java String Pool, Why Java String is Thread-safe?

- Homework: SOLID

## Sprint 3:

- Design Pattern Introduction Singleton + Factory Design Pattern by Example. - Rakib - 1 aug

- Java Generics, Type Infer, Wildcard, PECS, Upper Bound and Lower Bound - Iabur

- Collection Framework- me

- Java Concurrency Threading Model - Asif

- Homework 1 : Executor Service, Java Atomic Reference - Nishad

- Homework 2 : OSI Model Study - Nishad

## Sprint 4:

- Client - Server Architecture. Socket, Port Process Mapping.

- TCP Protocol Explanation.

- Implement a Java Socket Base Application.

- What happens when you type google.com on your browser

- Homework 1 : Use Multithreading in our Existing Socket-Based Application

- Homework 2 : maven build tool.

## Sprint 5:

- What is a Build Tool? Why Maven. Maven Installation.

- Java Web Servlet and Tomcat Server

- How Spring Application Context Works

- What is Spring Beans What is IoC

- Difference between Spring MVC & Spring Boot

- Basic REST Api Example with Spring Boot + Jar Deployment

- Homework 1 : What is Containerisation ? Different Linux Namespaces. Github Reference on Container Networking

- Homework 2 : Host to Host Communication

## Sprint 6:

- What is a container, why do we need this?

- Bare Metal vs VM vs Container.

- How Docker Works? Docker CLI, Docker Daemon, Docker Hub.

- How does Docker Networking work?

- Dockerize Spring Boot Application using Dockerfile and Docker Compose

- Homework: ACID property

## Sprint 7:

- What is ORM, Why do we need it ?

- JPA, Hibernate ORM Spring Data JPA (Java Reflection API) Docker compose bridge networking with DB Container

- Docker Volume Mapping, How to make DB container Persistent

- Spring Data JPA Integration with our Basic Rest Api

- Homework: Develop a User Management REST API Service

## Sprint 8:

- Application Scaling, Vertical vs Horizontal Scaling.

- What is Load Balancer ? Why do we need this?

- Single point of failure

- Active Active vs Active Passive Load balancing

- Load Balance to our Application using Nginx

## Sprint 9:

- Database scaling. Vertical vs Horizontal

- DB Master-Slave Architecture.

- Handling Read Heavy System Design with Horizontal DB Scaling with ProxySQL

- Homework: Implementation + Github Doc Reference

## Sprint 10:

- What is caching ? When do we need this ?

- Comparison between Different caching strategies. Cache Eviction Policy

- Data Caching with Redis

- Redis integration with Spring Boot

- Homework: Dockerize your Redis Server

## Sprint 11:

- Microservice Architecture

- 12 Factors of Microservice

- API Gateway,

- Service Discovery

- Basic REST api Communication

- Distributed Tracing using Zipkin

- Homework: Dockerized Microservices

## Spring 12:

- Spring AOP, Global Exception Handling with AOP.

- Global Exception Handling using Controller Advice

- Application Logging using Spring Logback.

- Centralized Logging using ELK Stack

- Homework: Basic Cloud Networking, Overlay Networking

## Sprint 13:

- Basic Cloud Networking Architecture Recap

- AWS Networking Infrastructure Readiness for K8s

- Setup Kubernetes Cluster (Hands-on Bare-metal)

- Deploying Nginx with Nodeport Service (Hands-on)

- Homework: Deploy Microservice Application in Kubernetes

## Sprint 14:

- What message queue, why do we need this?

- Notification service design with RabbitMQ

- RabbitMQ Integration with Spring Boot (Hands-on)

## Sprint 15:

- What is kafka? Why do we need this

- Data migration using Kafka Connect

- Kafka Integration with Spring Boot (Hands-on)

—--------------------------------------------------------------------------------------------------