|  |  |
| --- | --- |
| **Day** | **Module Name** |
| 1 | **Java 8**   * Introduction to Java * Introduction of IDE (Net Beans or Eclipse). How to configure and create java application with IDE. * Java in Server Environments * The Java Community Process * Java Fundamental * Overloading methods * Variable argument methods * Constructor overloading * Access modifiers: private, protected and public   **String processing**   * String manipulation with StringBuilder and StringBuffer * Essential String methods * Text parsing in Java * Input processing with Scanner * Text output and formatting * Regular expressions with the Pattern and Matcher classes |
| 2 | * Inheritance and Polymorphism * Sub-classing * Classes in Java * Polymorphism * Method overriding * The instanceof operator * Casting object references * Overriding Object methods * Final method and class   **Advanced Class Design**   * Abstract classes and type generalization * The static and final modifiers * Field modifier best practices * Designing abstract classes * Nested classes * Anonymous Class * Annotations   **Inheritance with Java Interfaces**   * Object composition and method delegation * Implementing multiple interfaces * Default method and static method |
| 3 | **Exceptions and Assertions**   * Best practices using exceptions * Assertions   **I/O Fundamentals**   * I/O using Java * Reading the console input stream * Writing to the console * Using I/O Streams * Reading and writing objects using Serialization |
| 4 | **Generics and Collections**   * Generic classes and type parameters * Type inference (diamond) * Collections and generics * List, set and Map * Stack and Deque * Comparable and Comparator * Functional Interface (Predicate, Consumer etc.) * Lamda Expression * Streams (Filter, map, foEach collectors etc..) |
| 5 | **Threading**   * Operating system task scheduling * Recognizing multithreaded environments * Creating multi-threaded solutions * Synchronization and Deadlock * ThreadPool   **Unit Testing**   * Maven Introduction * Introduction to JUNIT * Why testing? |
| 5 | * Junit with Eclipse * Assert methods and Annotation * Parameterized Test * Test Suite * Mockito * GIT (Self Study) * Assessment ( 45 Minutes MCQ + 1h 30m Code challenges) |
| 6 | **Design Pattern**  **Creational Patterns**   * Singleton * Factory * Abstract Factory * Prototype * Builder   **Structural Patterns**   * Adapter * Proxy * Façade   SOLID |
| 7 | Project |
| (Self Work) | Project |
| 8, 9 | **RDBMS**   * Database * Table (create, alter, delete) * Column constraints * Schema * DDL statements * DML statements, where clause, and/or, like, group by & having, foreign key, subquery, JDBC Driver * Statement/Prepared/Callable Statements |
| (Self Study) | **MongoDB**   * How MongoDB is different from RDBMS? * Need for NoSQL databases, & Terminology Collection * Document, |
| (Self Study) | * Create/Drop Database & Collection, insert/query/update/delete * Project/limiting/sorting/indexing * Aggregation/replication/sharding |
| 10 | **Spring 4**  **Spring Essentials**   * Why and what is Spring * Inversion of Control and Dependency Injection * Introduction to Bean Factory * Building your first Spring application   **Spring Container Concepts**   * Bean factory * Application context * Bean wiring * Managing bean lifecycle |
| 11 | **Spring Boot**   * What is Spring Boot * How Spring Boot works * Developing Web application using Spring Boot * Spring Boot integration with Spring Data * Auto configuration * Starter packages * Dependency management |
| 12 | **Spring Data**   * What is ORM? * Spring Boot Data/Jpa * CrudRepository, JpaRepository * Different Mappings, Custom Queries * MongoRepository * MongoTemplate |
| 13 | **Spring Rest**   * SOA Architecture overview * Spring MVC REST workflow * Spring REST introduction * Lifecycle of request in spring REST * Why REST Controller? * HTTP methods in REST * HTTP status code * HTTP Request Mapping * Restful URLs * Annotations in REST * ResponseEntity Object * Cross-origin Resource Sharing (CORS) * Rest Testing * Exception handling in REST at controller level and application level |
| 14, 15 | **Microservices**   * **Micro services Intro** * Monolith Service * Why Micro Service * Micro services Pros and Cons * Challenges in Micro Service * **Spring Boot Application** * Spring Boot enable Cloud Native * Introduction to Spring Rest Template / Asyc * Feign Client * How to implement client-side load balancing * How to implement a Naming Server (Eureka Naming Server) * How to connect the micro services with the Naming Server * Deploying spring application on cloud using Eureka * Configuring multiple spring applications to run on different port on cloud * Enabling Eureka server * Enabling Eureka client * Accessing Microservices through RestTemplate * Api Gateway |
| 16, 17 | Assessment ( 45 Minutes MCQ + 1h 30m Code challenges)  **Message Queue**   * Need for Message Queue * Producer, Consumer * Streaming - Pulsar Message Queue * Spring Boot Messaging (Publisher & Subscriber) |
| 18 | **Project Sprint 2** |