**DSA**

* Collections

1. List
2. ArrayList
3. LinkedList
4. Vector
5. Set
6. HashSet
7. LinkedHashSet
8. TreeSet
9. Map
10. HashMap
11. LinkedHashMap
12. TreeMap
13. HashTable
14. ConcurrentHashMap

**Java-8:**

* Streams
* Lambda Expressions
* Functional Interface
* Static Method
* Default method
* Predicate & Supplier and Consumer Interface
* Optional class

**Spring**: Student Application

* IOC container
* Dependency Injection
* Bean
* Autowiring
* Qualifiers
* Annotations
* DTOs – data transfer object
* Spring boot crud operations - POST, GET, PUT, DELETE Mappings
* Entity designing
* **Hibernate mappings** – oneToOne, manyToOne, OneToMany, ManyToMany
* **Jpa repository** --- Java Persistance API
* Jpa anotations
* Jpa queries – inbuilt, customised
* Jpql queries
* SQL -native queries
* Db connections with mysql
* Spring Profiles
* Actuators
* Spring boot Exception’s handlings – custom exceptions and global exceptions
* POST MAN
* GIT version control tool
* Lombok plugin
* Maven build tool
* Tomcat server
* **Microserices**:
* Spring cloud API Gateway
* Eureka server-client
* Rest Template/Feign client
* Resillaince4j

Springboot application:

**Bank application with following APIs**

* **Create Account**
* **Update User Customer details🡪**
* **Delete User Account details**
* **Deactivate Account**
* **Fetch Account info by accountNumber 🡪**
* **Fetch all accounts existing for particular Customer**
* **Fetch only active accounts existing for particular Customer**
* **Fetch all active customers details**
* **Fetch Customer details with customerId**
* **Fetch account balance**
* **Transfer amount from one account to another account**
* **Deposit/withdraw the amount**

Microservices application:

**E-commerce application with 2 services calling APIs internally**

Product service

Cart Service

Api gateway

Eureka service registry