

## **MODULE 1. Java SE**

### **Lesson 1. Introduction to programming**

- i¶ Introduction to programming
- i¶ Algorithmic thinking, Reasoning
- i¶ Flowcharts, Pseudo-codes
- i¶ Console and Desktop Applications.
- i¶ Environment setup.
- i¶ Introduction to IDE editor and tools.
- i¶ Introduction to execution of java programs.
- i¶ Java compiler. JIT, JVM, JRE, JDK
- i¶ javac vs. java commands.
- i¶ .java vs .class extensions.
- i¶ Writing Simple Hello World java program.
- i¶ Main entry point and command line arguments.
- i¶ Main shell commands for compiling and running.
- i¶ Java syntax

### **Lesson 2. Language Elements**

- i¶ Comments, Variables and Data types
- i¶ How memory works for the variables of primitive data type. Final variables.
- i¶ Expressions and statements.
- i¶ What are operators.
- i¶ How many types of operators are there in java
- i¶ Unary Operator, Arithmetic Operator, Shift Operator, Relational Operator
- i¶ Bitwise Operator, Logical Operator, Ternary Operator and Assignment Operator
- i¶ Input and output process in Java.
- i¶ What is Input and Output.
- i¶ Input types (Scanner), output formats.

### **Lesson 3. Conditions**

- i¶ if/else operator
- i¶ switch/case operator
- i¶ ternary if operator
- i¶ Using binary operators and Boolean data types

### **Lesson 4. Repetitions**

- i¶ Introduction to loops
- i¶ While loop
- i¶ do/while loop
- i¶ for loops.
- i¶ break, continue

### **Lesson 5. Arrays**

- i¶ Declaration, instantiation, initialization of arrays
- i¶ How memory works for arrays. Java and Spring Boot Microservices Full Course
- i¶ Single dimensional.
- i¶ Iterating through an array. For-each statement.

## **Lesson 6. Introduction to Object Oriented Programming**

- ï¶ Classes vs Objects
- ï¶ Primitive vs Reference data type
- ï¶ How memory works for reference data types
- ï¶ Garbage collection.
- ï¶ Methods in Java Types of methods.
- ï¶ Defining and calling methods, void keyword

## **Lesson 7. Introduction to Object Oriented Programming**

- ï¶ Constructors
- ï¶ Static and instance methods
- ï¶ Method overloading
- ï¶ Object initialization
- ï¶ Pass-by-value vs pass-by-reference
- ï¶ Sending and returning arrays to/from a method

## **Lesson 8. OOP Principles**

- ï¶ Encapsulation
- ï¶ Inheritance
- ï¶ Method overriding

## **Lesson 9. OOP Principles (continues)**

- ï¶ Polymorphism
- ï¶ Abstraction
- ï¶ `instanceof` keyword

## **Lesson 10. Interfaces**

- ï¶ Interfaces and implementing methods
- ï¶ Default and static interface methods
- ï¶ Design Patterns. Singleton pattern, builder pattern, factory pattern
- ï¶ **In-class quiz**

## **Lesson 11. Packages**

- ï¶ Packaging. Built-in packages
- ï¶ Importing packages. Single class imports
- ï¶ Static imports, Whole package imports (using wildcards)
- ï¶ User defined packages.
- ï¶ Final methods, final classes, polymorphism, abstract methods
- ï¶ UML Diagrams for class designing
- ï¶ Encapsulation (different packages: default vs protected)

## **Lesson 12. Enumerations and Wrapper types**

- ï¶ Enumerations
  - ï¶ Methods, calling other methods and passing variables
  - ï¶ local variables, formal parameters
- Java and Spring Boot Microservices Full Course
- ï¶ Passing by value.
  - ï¶ Wrapper alternatives of primitive types like Integer, Long etcâ€¦
  - ï¶ Var keyword

## **Lesson 13. Exceptions in Java**

- ï¶ Exception hierarchy
- ï¶ checked vs unchecked exceptions

- ï¶ Compile-time vs run-time
- ï¶ Errors, Handling exceptions
- ï¶ try, catch, finally blocks
- ï¶ Multiple catch vs union catch
- ï¶ throws statement
- ï¶ Try-with-resources
- ï¶ Throwing new exceptions
- ï¶ Custom exceptions
- ï¶ Swallowing exceptions

## **Lesson 14. Date and Time API & Multidimensional arrays**

- ï¶ LocalDate
- ï¶ LocalDateTime
- ï¶ LocalTime
- ï¶ Instant
- ï¶ Period
- ï¶ ChronoUnit
- ï¶ Multidimensional and jagged arrays.
- ï¶ Array Copy and Array Clone.
- ï¶ Finding maximum/minimum of an array.
- ï¶ Sorting arrays.
- ï¶ Introduction to data structures and algorithms: Bubble sort, linearsearch, selection sort, binary search.

## **Lesson 15. Generics**

- ï¶ Need for generics, Type wildcards
- ï¶ Diamond operator
- ï¶ Generic class definitions
- ï¶ Generic method definitions

## **Lesson 16. Sorting and Comparing**

- ï¶ Sorting collections.
- ï¶ Comparator, Comparable class.
- ï¶ Properties class.
- ï¶ Reading properties from external files.

## **Lesson 17. Collections**

- ï¶ Intro to java collection framework
- ï¶ Data structures: ArrayList, LinkedList, Map, Set, Vector, Stack, Queue.Java and Spring Boot Microservices Full Course
- ï¶ Ready JDK implementations.
- ï¶ List interface and its classes, set interface and its classes
- ï¶ Map interface and its classes

## **Lesson 18. Collection implementations**

- ï¶ Set vs List.
- ï¶ LinkedHashSet, TreeSet
- ï¶ PriorityQueue
- ï¶ EnumSet class, EnumMap class
- ï¶ Collections class, Arrays class

## **Lesson 19. Input Output Streams & Reading and Writing files**

- 📖 Standard Streams
- 📖 Input, Output and Error
- 📖 Byte and Character IO Streams
- 📖 Several Byte Stream classes
- 📖 Several Character Stream classes
- 📖 Files and IO
- 📖 FileReader and FileWriter
- 📖 File navigations
- 📖 Buffered byte and character streams

## **Lesson 20. Serialization, Reflection**

- 📖 Serialization, Object Streams
- 📖 Transient keyword
- 📖 Binary vs XML vs JSON serialization
- 📖 Introduction to Reflection API
- 📖 Java Class object, Fields, Methods and Constructors
- 📖 Private vs public modifiers
- 📖 Accessing inherited fields and methods
- 📖 Dynamic invocation, Annotations, Arrays, Generics

## **Lesson 21. Multithreading**

- 📖 Multithreading, Process vs Thread vs Task
- 📖 Thread class and Runnable interface
- 📖 Lifecycle of a thread, Synchronization
- 📖 **In-class quiz**

## **Lesson 22. Execution Service**

- 📖 Execution service
- 📖 Concurrency
- 📖 Atomic scalars
- 📖 JoinFork

## **Lesson 23. Data Structures and Algorithms. Lambda. Stream Api**

- 📖 Intro to DSA
- 📖 Complexity analysisJava and Spring Boot Microservices Full Course
- 📖 Big Oh notation
- 📖 Functional interface
- 📖 Lambda expressions.
- 📖 Java Stream API and lambda expressions to search collections

## **MODULE II. Introduction to Databases**

### **Lesson 24. Database fundamentals.**

- 📖 Introduction to RDBMS, SQL commands
- 📖 DML, DDL, TCL, DCL
- 📖 Writing SQL Statements
- 📖 Adding a New Row to a Table
- 📖 The INSERT Statement Syntax 8-5
- 📖 Inserting New Rows
- 📖 Creating various database objects, and viewing their list with special SELECT statements.

## **Lesson 25. Retrieving, Restricting and Sorting Data**

- i¶ Arithmetic Expressions and Operators
- i¶ Null Values in Arithmetic Expressions
- i¶ Defining a Column Alias, Concatenation
- i¶ Using Operators
- i¶ Literal Character Strings
- i¶ Duplicate Rows, Eliminating Duplicate Rows
- i¶ Limiting Rows Using a Selection
- i¶ Using the WHERE Clause
- i¶ Character Strings and Dates
- i¶ Using Comparison Conditions " BETWEEN, IN, LIKE
- i¶ Using Logical Conditions " AND, OR, NOT
- i¶ ORDER BY Clause
- i¶ Sorting by Column Alias, Multiple Columns
- i¶ GREATEST, LEAST, NULLIF, COALESCE, CASE/WHEN Functions
- i¶ Conditional Expressions - CASE and DECODE Function
- i¶ Using TO\_CHAR, TO\_NUMBER, TO\_DATE, CAST, "INTERVAL Functions; Date Formatting
- i¶ Character Manipulation, Number, Round, Trunc, Mod, Conversion Functions

## **Lesson 26. Aggregating Data Using Group Functions. Subqueries Objectives**

- i¶ What Are Group Functions?
- i¶ Types of Group Functions
- i¶ Group Functions SyntaxJava and Spring Boot Microservices Full Course
- i¶ Using the AVG and SUM Functions
- i¶ Using the MIN and MAX Functions
- i¶ Using a Subquery to Solve a Problem
- i¶ Subquery Syntax
- i¶ Using a Subquery
- i¶ Using SET Operators

## **Lesson 27. Constraints. Displaying Data from Multiple Tables**

- i¶ Obtaining Data from Multiple Tables.
- i¶ Cartesian Products
- i¶ Generating a Cartesian Product
- i¶ Types of Joins
- i¶ Primary key, Foreign Key, Unique, Not Null, Check

## **Lesson 28. Java Database Connectivity**

- i¶ JDBC API
- i¶ Java Database Drivers
- i¶ Connection to database
- i¶ Statement, Callable Statement, Prepared Statement
- i¶ Result Sets
- i¶ Auto commit

## **Midterm Exam Week**

## MODULE III – JAVA EE

### Lesson 29. Introduction to Spring Boot Application

- Server-Side Rendering
- What is Web API
- HTTP protocol. Status Codes and HTTP Methods
- JSON and YAML formats
- IoC, DI
- Spring Initializr. start.spring.io
- Spring Framework Overview. ApplicationContext
- Spring Boot and Related Projects Overview
- Creating first Spring Application
- Build tools: Gradle, Maven
- Bean configuration types
- @Configuration, @Bean

### Lesson 30. Simple Spring Boot ApplicationJava and Spring Boot Microservices Full Course

- @SpringBootApplication annotation. Conventional projectstructure.
- @Controller, @RequestMapping, @GetMapping, @PostMapping and other request mapping annotations
- @RequestBody, @ResponseBody, @RestController
- @RequestHeader, @CookieValue
- HttpServletRequest, HttpServletResponse
- Handling query parameters from query string, path
- ResponseEntity
- @ResponseStatus
- Postman
- Saving submitted model data to a List and manipulating that list with various requests

### Lesson 31. Working with Services and Configuration

- Creating services and injecting with @Autowired
- Injecting with constructor and setters.
- application.properties, application.yaml files.
- Specifying environment variables in application properties
- Using @Value annotation.
- Specifying property values with ENV VARS and default values
- Overriding properties from CLI during java -jar command

### IN-class Quiz

### Lesson 32. Data Layer. JDBC Template. Profiles. Lombok

- @Repository.
- Injecting different implementations of the services based on the @Profile
- Specifying profiles in IDE and CLI
- JDBC Template. CRUD methods.
- Adding Lombok dependency.
- Using Lombok to create models easily
- Lombok annotations: @Data, @Getter @Setter annotation
- @Builder and other annotations
- No Argument Constructor

ï¶ All Argument Constructor

### **Lesson 33. Logging. Swagger**

ï¶ What is log?

ï¶ Log levels

ï¶ Simple Logging with LoggerFactory.getLogger()

ï¶ @Slf4j annotation and default logging features

ï¶ Logback: console, json

ï¶ Logging levels

ï¶ Logging format

ï¶ File Output

ï¶ Log Groups

ï¶ Logging properties and customizationJava and Spring Boot Microservices

Full Course

ï¶ Profile specific logging

ï¶ What is OpenAPI Specification.

ï¶ Overview of Swagger and SpringFox libraries

ï¶ Adding swagger and SpringFox dependencies

ï¶ @Configuration annotation

ï¶ Basic Swagger Configuration

ï¶ Calling application methods through Swagger UI

ï¶ Changing default swagger URL

ï¶ Documenting models with swagger annotations

ï¶ server.servlet.context-path property

### **Lesson 34. Rest Controller Advice. Multipart.**

ï¶ Throwing and Handling Exceptions

ï¶ @ExceptionHandler annotation

ï¶ Rest Controller Advice

ï¶ Adding necessary headers and HTTP statuses during exceptions

ï¶ Show exception messages in responses server.error.include-  
message=always

server.error.include-stacktrace=never

ï¶ Uploading files

ï¶ Saving uploaded files

ï¶ Downloading files

ï¶ Uploading files with additional JSON models

### **Lesson 35. Adding Database Support**

ï¶ Spring Data Overview

ï¶ JPA and Hibernate Overview

ï¶ Adding Database Driver Dependencies

ï¶ Specifying Database Connectivity Parameters in properties file

ï¶ Creating @Entity models

ï¶ Creating @Repositories with CrudRepository and JpaRepository

ï¶ Injecting repositories into Service implementations

ï¶ Creating objects in database. Updating, Searching and Deleting

ï¶ @Table, @Column, @Id and @GeneratedValue annotations

### **Lesson 36. More about JPA**

ï¶ @OneToOne relationship and its relationship

- ï¶ @OneToMany annotation and its relationship
- ï¶ @ManyToMany annotation and its relationships
- ï¶ Custom Repository Functions
- ï¶ @Queries with HQL
- ï¶ Transactions
- ï¶ Committing and Rolling Back

### **Lesson 37. JPA Repositories Continued. MapStruct bean mappings**

- ï¶ @Native queries. Building complex queries
- ï¶ @EntityGraph annotation.
- ï¶ Hibernate related parameters in application.properties
- ï¶ Connection PoolsJava and Spring Boot Microservices Full Course
- ï¶ Mapper overview
- ï¶ Adding dependencies
- ï¶ @Mapper interface
- ï¶ @Mapping interface
- ï¶ Obtaining an instance of mapper at the service
- ï¶ Testing MappersJava and Spring Boot Microservices Full Course

### **Lesson 38. Database Structure Versioning â€” Liquibase**

- ï¶ Liquibase overview
- ï¶ Adding Liquibase dependencies
- ï¶ Liquibase and Lombok. Maven build plugins.
- ï¶ Creating migration files to create tables
- ï¶ Seeding data into tables
- ï¶ Create master migration file and importing child versions
- ï¶ Manually modifying existing migration file and troubleshooting database change version

### **Lesson 39. Docker Containers**

- ï¶ Virtualization and containers
- ï¶ Installing Docker
- ï¶ Docker images and containers
- ï¶ Pulling and running images
- ï¶ Running commands inside docker containers
- ï¶ Watching logs of containers
- ï¶ Running PostgreSQL database from docker
- ï¶ Running RabbitMQ and Redis images from docker
- ï¶ Dockerfile overview
- ï¶ Base images.
- ï¶ Dockerfile commands.
- ï¶ Adding files, executing commands
- ï¶ Specifying entry point
- ï¶ Build Spring Boot application docker image
- ï¶ Run multiple Spring Boot images

### **Lesson 40. Spring Validation**

- ï¶ Spring Validation overview. Spring Validation
- ï¶ Adding dependencies to pom.xml
- ï¶ @Min, @Max, @Size, @Email annotations



- @NotEmpty, @NotNull, @NotBlank annotations
- @Positive, @Negative, @Pattern and other annotations
- @Valid and @Validated annotations
- Custom validators
- Validating programmatically

### • In-class Quiz

## **Lesson 41. Internationalization, Static Files and Resources**

- Serving static content
  - Serving HTML, CSS, and images resources
  - Locations of static files
  - i18n and message.properties
  - LocaleResolver and LocaleChangeInterceptorJava and Spring Boot
- Microservices Full Course

## **Lesson 42. Spring Web - Thymeleaf**

- Adding Thymeleaf dependency
- @Controller annotation and returning views
- Thymeleaf markup tags overview
- Passing model to Thymeleaf templates
- Building forms and handling POST data

## **Lesson 43. Spring Security**

- Authentication, Authorization
- JWT Generation
- Database Backed Users Service
- Security Configuration
- Enabling and Disabling Requests to certain paths
- Password Hashing
- Filters and Interceptors
- @CrossOrigin annotation
- Allowing Authentication Bearer header in Swagger

## **Lesson 44. Spring Security Continued**

- More about JWT. jwt.io
- Hashing algorithms. MD5, SHA1, SHA256
- Signing and verifying signature.
- Cryptography. RSA, DES, AES and others
- X.509 certificates
- Nimbus library

## **Lesson 45. Scheduled Jobs**

- Scheduled Jobs overview.
- Creating jobs.
- @Scheduled annotation
- @EnableScheduling annotation
- fixedRate, fixedDelay and initialDelay parameters
- Using Cron Expressions

## **Lesson 46. Making HTTP Calls to other systems**

- Working with RestTemplate to retrieve various resources
- Feign Client

- ï¶ Using various HTTP Methods to retrieve text, objects and binary data
- ï¶ Specifying headers
- ï¶ Configuring Timeouts
- ï¶ JSON serialization and deserialization

### **Lesson 47. SOAP Web Services**

- ï¶ Creating SOAP Web Services with Spring
- ï¶ JAX-WS SOAP Web Service Client
- ï¶ Contract first/last servicesJava and Spring Boot Microservices Full Course
- ï¶ WSDL, XSD definitions
- ï¶ JAXB XML serialization and deserialization
- ï¶ Reading and Writing XML Documents

### **Lesson 48. Microservices Architecture**

- ï¶ Overview of Microservices
- ï¶ Monolith vs Microservices
- ï¶ Microservices Patterns
- ï¶ Avoid Binary Dependency
- ï¶ Docker-compose multiple microservices

### **Lesson 49. More about Microservices**

- ï¶ API Gateway
- ï¶ Database per Service pattern
- ï¶ SAGA pattern
- ï¶ Circuit Breaker pattern
- ï¶ Sidecar pattern
- ï¶ Service mesh
- ï¶ Health checks

### **Lesson 50. Testing Microservices**

- ï¶ Unit testing
- ï¶ Integration Testing:
  - i. Consumer-driven contract test
  - ii. End-to-end testing
  - iii. Mocking
- ï¶ Building integration test with a @Service implementation that is mocking functionality when launched with special @Profile

### **Lesson 51. Git version control system**

- ï¶ Overview about version controlsystems
- ï¶ Branches, Commits, Releases
- ï¶ Initializing or cloning a repository
- ï¶ Listing branches
- ï¶ Git status
- ï¶ Adding changes
- ï¶ Commit-in changes
- ï¶ Pushing, pulling, fetching
- ï¶ Working with branches

- ¶ Merging branches
- ¶ Git servers. Gitlab, Github, Bitbucket
- ¶ Resolving conflicts
- ¶ Rebasing Java and Spring Boot Microservices Full Course

## ***Lesson 52. Continuous Integration and Continuous Delivery***

- ¶ Overview of CI/CD
- ¶ What is pipeline
- ¶ Run Gitlab server from docker image
- ¶ Register runners
- ¶ Gitlab pipeline example.
- ¶ Specifying pipeline variables
- ¶ Overview of Jenkins, Github Actions and CircleCI.

## ***Lesson 53. Deploying Microservices to Kubernetes***

- ¶ Overview of Kubernetes
- ¶ Setting up Minikube.
- ¶ Nodes, Pods, Deployments
- ¶ Replication Controller, ReplicaSets
- ¶ Deploying to Kubernetes

## ***Lesson 54. Accessing Microservices inside of Kubernetes***

- ¶ Exposing deployments with Services
- ¶ Service Types: NodePort, LoadBalancer and ClusterIP
- ¶ Ingress Controller and Ingress Resources
- ¶ HAProxy configuration for routing resources to ingress controller
- ¶ Health checks. Readiness and Liveness probes
- ¶ Adding Kubernetes support to CI/CD

## ***Lesson 55. Monitoring Microservices and Log collection***

- ¶ Monitoring overview
- ¶ Prometheus and Grafana
- ¶ Collecting metrics with Spring
- ¶ Collecting logs from containers with Loki
- ¶ Building dashboards and alarms