Java and Spring Boot Microservices Full Course

COURSE SYLLABUS

1+1 Model:
55 theoretical lessons
55 practical lessons
(Each theoretical session has its practical session)

Each session: 2 hours
Course duration: 7 months (+ 1 month for project)

COURSE OUTCOMES

1. LEARNING JAVA PROGRAMMING LANGUAGE AND OBJECT ORIENTED PROGRAMMING
2. IDENTIFY THE APPROPRIATE DATA STRUCTURES AND ALGORITHMS FOR SOLVING REAL WORLD PROBLEMS.
3. PREPARING FOR THE ORACLE INTERNATIONAL CERTIFICATION EXAM
4. DESIGNING DATABASE SCHEMAS AND CONNECT TO A DATABASE
5. USING PROGRAMMING TOOLS LIKE AN IDE, GIT, FRAMEWORKS LIKE SPRING
6. ENSURING THE BASIC SECURITY OF A WEB APPLICATION
7. TESTING YOUR APPLICATIONS USING TEST DRIVEN DEVELOPMENT
8. DEVELOPING A FULLY FUNCTIONING WEBSITE AND DEPLOY ON A WEB SERVER
9. UNDERSTANDING HOW KUBERNETES CAN BE USED TO DEPLOY MICROSERVICES

Grading Policy:

Grade	Final score	Honor degree
A	85 – 100	High
B	70 – 84	Merit
C	50 – 69	Pass
F	< 50	Fail

Assessment:

Graded Activity	Weight
Midterm exam (theoretical and practical)	30%
Student assessment	10%
Final Project	30%
Final exam	30%

1. **Midterm exam:**

Midterm exam is held 1 time. Exam questions cover all topics on **Module I and Module II.**

Assessment type	Format
Theoretical	Multiple choice or open-ended
Practical	In-Class or task based

2. **Student assessment:**

Student assessment is the process of evaluating students' abilities and achievements. Each student is evaluated every month during their education period by the following criteria:

Student's performance:

- Takes an active part in classes
- Listens carefully to lessons
- Asks questions when he doesn't understand the lesson
- Motivation is high

Student's technical knowledge:

- Performs and understands assigned tasks
- He can apply what he knows and has some knowledge
- Can write code without any help

3. **Final Project:**

The final project enables to show student's comprehension of the course topics, and capabilities in problem formulation, presentation, and team working. Projects are assigned by ATL Academy or ATL Tech.

How you will be graded:

- Completion of task
- Code review/structure
- Content of knowledge

4. **Final exam:**

The final exam will be based on theoretical questions. Exam questions consist of all topics delivered during whole session.

Assessment type	Format
Theoretical	Multiple choice or open-ended

Contents

MODULE 1. Java SE

Lesson 1. Introduction to programming

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

Lesson 2. Language Elements

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

Lesson 3. Conditions

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

Lesson 4. Repetitions

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

Lesson 5. Arrays

- Declaration, instantiation, initialization of arrays
- How memory works for arrays.

- Single dimensional.
- Iterating through an array. For-each statement.

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

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

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

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

Lesson 8. **OOP Principles**

- Encapsulation
- Inheritance
- Method overriding

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

- Polymorphism
- Abstraction
- “instanceof” keyword

Lesson 10. **Interfaces**

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

Lesson 11. **Packages**

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

Lesson 12. **Enumerations and Wrapper types**

- Enumerations
- Methods, calling other methods and passing variables
- local variables, formal parameters

- Passing by value.
- Wrapper alternatives of primitive types like Integer, Long etc…
- Var keyword

Lesson 13. **Exceptions in Java**

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

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

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

Lesson 15. **Generics**

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

Lesson 16. **Sorting and Comparing**

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

Lesson 17. **Collections**

- Intro to java collection framework
- Data structures: ArrayList, LinkedList, Map, Set, Vector, Stack, Queue.

- Ready JDK implementations.
- List interface and its classes, set interface and its classes
- Map interface and its classes

Lesson 18. **Collection implementations**

- Set vs List.
- LinkedHashSet, TreeSet
- PriorityQueue
- EnumSet class, EnumMap class
- 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 analysis

- 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**

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

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

Objectives

- What Are Group Functions?
- Types of Group Functions
- Group Functions Syntax

- ❖ Using the AVG and SUM Functions
- ❖ Using the MIN and MAX Functions
- ❖ Using a Subquery to Solve a Problem
- ❖ Subquery Syntax
- ❖ Using a Subquery
- ❖ Using SET Operators

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

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

Lesson 28. **Java Database Connectivity**

- ❖ JDBC API
- ❖ Java Database Drivers
- ❖ Connection to database
- ❖ Statement, Callable Statement, Prepared Statement
- ❖ Result Sets
- ❖ 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 Application**

- ❖ @SpringBootApplication annotation. Conventional project structure.
- ❖ @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
- ❖ All Argument Constructor

Lesson 33. **Logging. Swagger**

- ❖ What is log?
- ❖ Log levels
- ❖ Simple Logging with LoggerFactory.getLogger()
- ❖ @Slf4j annotation and default logging features
- ❖ Logback: console, json
- ❖ Logging levels
- ❖ Logging format
- ❖ File Output
- ❖ Log Groups
- ❖ Logging properties and customization

- Profile specific logging
- What is OpenAPI Specification.
- Overview of Swagger and SpringFox libraries
- Adding swagger and SpringFox dependencies
- @Configuration annotation
- Basic Swagger Configuration
- Calling application methods through Swagger UI
- Changing default swagger URL
- Documenting models with swagger annotations
- server.servlet.context-path property

Lesson 34. Rest Controller Advice. Multipart.

- Throwing and Handling Exceptions
- @ExceptionHandler annotation
- Rest Controller Advice
- Adding necessary headers and HTTP statuses during exceptions
- Show exception messages in responses server.error.include-message=always server.error.include-stacktrace=never
- Uploading files
- Saving uploaded files
- Downloading files
- Uploading files with additional JSON models

Lesson 35. Adding Database Support

- Spring Data Overview
- JPA and Hibernate Overview
- Adding Database Driver Dependencies
- Specifying Database Connectivity Parameters in properties file
- Creating @Entity models
- Creating @Repositories with CrudRepository and JpaRepository
- Injecting repositories into Service implementations
- Creating objects in database. Updating, Searching and Deleting
- @Table, @Column, @Id and @GeneratedValue annotations

Lesson 36. More about JPA

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

Lesson 37. JPA Repositories Continued. MapStruct bean mappings

- @Native queries. Building complex queries
- @EntityGraph annotation.
- Hibernate related parameters in application.properties
- Connection Pools

- Mapper overview
- Adding dependencies
- @Mapper interface
- @Mapping interface
- Obtaining an instance of mapper at the service
- Testing Mappers

Lesson 38. Database Structure Versioning – Liquibase

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

Lesson 39. Docker Containers

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

Lesson 40. Spring Validation

- Spring Validation overview. Spring Validation
- Adding dependencies to pom.xml
- @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 LocaleChangeInterceptor

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

Lesson 47. **SOAP Web Services**

- Creating SOAP Web Services with Spring
- JAX-WS SOAP Web Service Client
- Contract first/last services

- WSDL, XSD definitions
- JAXB XML serialization and deserialization
- Reading and Writing XML Documents

Lesson 48. **Microservices Architecture**

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

Lesson 49. **More about Microservices**

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

Lesson 50. **Testing Microservices**

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

Lesson 51. **Git version control system**

- Overview about version control systems
- Branches, Commits, Releases
- Initializing or cloning a repository
- Listing branches
- Git status
- Adding changes
- Commit-in changes
- Pushing, pulling, fetching
- Working with branches
- Merging branches
- Git servers. Gitlab, Github, Bitbucket
- Resolving conflicts
- Rebasing

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

Final Project