# Backend Development with Java

*Detailed Course Syllabus*

| | |
|---|---|
| ***Instructor:*** | ***Elvin Taghizade ©*** |
| ***Duration:*** | ***6 months (can be divided into two parts)*** |
| ***Lesson time:*** | ***2 time a week*** |

## *A. First part - Programming basics with Java*

1. Introduction / Software Setup
    a. Introduction to ICT (Career plan)
        - Syllabus intro
        - Course procedures
        - Requirements & Questions
        - Basics of ICT
        - Information in digital form
        - Number systems and conversion among them
    b. What is programming?
        - Introduction to Programming (Roadmap)
    c. What are programming languages?
        - How to choose PL and identify proper roadmap
    d. What is VCS (Git / GitHub)?
        - Understanding of VCS
        - Git download and installing
    e. JDK / JRE / JVM / IDE?
        - How to run java code from console (terminal)
        - .java and .class (byte code) files
        - Compiling and running
        - Differences between JDK and JRE
        - What is JVM?
        - IDEs - **Intellij IDEA**, NetBeans, Eclipse
2. Java basics
    a. Java syntax (“Hello, World!” in Java)
    b. Application lifecycle: writing, compiling (to bytecode), running
    c. Application entry point:
    `public static void main (String[] args){}`
    d. Print to console
        - `System.out.println(“Hello, World”);`
        - `System.out.print(“Hello, World”);`
        - `System.out.printf(“Hello, %s”, “World”);`
    e. Java basics
        - Variables & Data types
    f. Operations
        - Arithmetic operations

- Relational operations
- Logical operations
- Assignment operations
- Miscellaneous operations

g. Practice: “Problem solving” (e-olymp)

3. Control Flow
   a. Input from keyboard/console - `Scanner` class
      - Difference between `next() & nextLine()`
   b. Conditional statements
      - `if`
      - `if - else`
      - `if - else if - else`
      - `switch - case`
   c. Loops
      - `for`
      - `while`
      - `do-while`
      - enhanced for loop - ‘foreach’ (after arrays)
   d. Break/continue statements
   e. Nested conditions and loops
   f. Practice (Problem Solving)
4. Methods
   a. What is a method in Java?
      - declaration and initialization
      - `static` and `non-static`
   b. Parameters
      - Parametric methods
      - Non-parametric methods
   c. What is Return type?
      - `void` methods
      - value methods
   d. Overloading
   e. Recursion (in algorithms sections)
   f. Practice: “Calculator app with switch and methods”
5. Arrays
   a. One dimensional arrays
      - declaration

        - initialization
    b. Operations on an array (fill, print, copy etc.)
    c. Two and more dimensional arrays
    d. Practice: Problem solving
6. String
    a. Recap
    b. Types of data types (primitive and non-primitive(reference))
    c. Char array and understanding `String`
    d. String under the hood (uses char array etc.)
    e. Methods of String class (some)
        - `toLowerCase() & toUpperCase()`
        - `substring() & trim()`
        - `indexOf(String s) & indexOf(int i)`
        - `split(), replace(), length(), concat()`
    f. Memory (RAM) intro
    g. Memory management in java
        - Stack
        - Heap
    h. Reference and how this works?
    i. Passing values
        - Passing-by-value
        - Passing-by-reference
    j. Memory management via diagrams - basic
        - String pool
    k. Memory management via diagrams - intermediate (method parameter)
    l. String concatenation:
        - “+” operator for strings
        - `concat()`
        - `StringBuilder`
        - `StringBuffer`
        - Comparison of above solutions
7. Practice
    a. General repetition of lessons learned
    b. Practice “Problem solving”
8. **First exam here**
    a. **Verbal**
    b. **Written (practical)**

9. Object-Oriented Programming (OOP) #1
    a. Object and class
    b. Constructors
    c. Access modifiers – Encapsulation
    d. Static and non-static difference
    e. Getters and setters
    f. `Class` loading
    g. Types of variables
        - Instance variables
        - Local variables
        - Static/global variables
    h. References/Garbage Collectors
10. Object-Oriented Programming (OOP) #2
    a. Inheritance and Polymorphism
    b. Keywords: `this & super`
    c. Overriding
    d. Compile-time and runtime polymorphism
11. Object-Oriented programming (OOP) #3
    a. Wrapper types
    b. Casting (Downcasting/Upcasting)
    c. Abstraction (`abstract`)
    d. Interfaces `(interface)`
12. Exceptions
    a. Error
    b. Exception
        - Checked and unchecked exceptions
    c. `Try-catch`
    d. `Try-catch-finally`
    e. `Try-with-resources (in files)`
    f. Own Exception
    g. Throw/throws
13. File Input-Output (I/O), Date & Time
    a. File reading and writing with “io”
    b. File reading and writing with “nio”
    c. `Date & Time` in Java
14. **Practice & Second exam: “1. In-Memory, 2. File managed app”**

## *B. Second Part – Algorithms and Data Structures****

1. Algorithms and Data Structures #1
    a. Overview
    b. Complexity notations
    c. Searching
        - Linear search
        - Binary search
    d. Sorting algorithms
        - Bubble sort
        - Insertion  sort
        - Selection  sort
        - Merge  sort
        - Quick sort
        - Heap sort
    e. Benchmarking & visualization
2. Algorithms and Data Structures #2
3. Collection API
    a. `ArrayList`
    b. `LinkedList`
    c. Hash implementation & hashing
    d. `Set`
    e. `Map`
    f. `Stack`
    g. `Queue`
    h. `Deque`

## *C. Third Part – Databases and SQL*

1. Database / SQL
    a. Intro to databases
    b. PostgreSQL / Oracle / MySQL / MS SQL / H2 overview
    c. Environment setup
    d. DDL / DML
    e. CRUD (create/retrieve(read)/update/delete) operations
    f. Practice: “Database operations”
2. Database #2
3. Database #3
4. Database #4
5. **Practice**
    a. **General repetition of lessons learned**
    b. **Practice “Problem solving”**
    c. **Exam**

## *D. Fourth Part – Java EE*

1. Web introduction
    a. How does a browser work?
    b. HTML & CSS & JavaScript
2. HTTP
    a. Http
    b. Request – Response
    c. Server
    d. Servlet
    e. Handler
    f. Mapping
3. Server and Servlet
    a. Jetty / Tomcat server
    b. Parameter parsing
    c. Server / Servlet lifecycle
4. Web Navigation and HTML Generation
    a. Static content (IMG, JS, CSS, etc.) & Freemarker
    b. File upload
5. Filters, Cookies and Basic Authorization (*optional*)
6. JSON, XML, REST, SOAP
7. Practice & Third exam:
8. “Clone web project”
9. Deployment.

## *E. Fifth Part – Spring Framework*

1. Spring intro
    a. Framework vs Library
    b. Lifecycle
    c. Project initializing
2. Spring Project
    a. Spring Core
    b. Spring MVC
    c. Spring Boot
    d. Minimal spring boot application
        - Port
        - Endpoint
3. Inversion of Control & Dependency Injection
    a. Setter injection
    b. Constructor injection
    c. @SpringBootApplication
    d. @Controller
    e. @RequestMapping
    f. @RequestBody
4. Thymeleaf, Project Lombok
    a. @Data
    b. @Value
    c. @AllArgsConstructor
    d. @NoArgConstructor
5. Consuming and serving REST
    a. @RequestParam
    b. @ResponseBody
    c. @PathVariable
    d. @RestController
    e. @Service
    f. @Repository
    g. @Autowired

6. DTO, Object mapping
7. Spring Data, JPA, ORM basics, relations, validations
8. Spring Security Architecture. Basic Auth (login+password)
9. Spring Security. Roles, Permissions, Configuration
10. Spring Security. Stateful
11. Spring Security. Stateless, Tokens. JWT token
12. Monolith vs Microservices
13. Deployment, Containers, Packing, Heroku
14. Final Lesson / Final Recap / Q&A session
15. Interview Preparation
16. Interview Preparation