



## **DTC-010 – Programming in Java**

# Programming in Java: Level 1

## OBJECTIVES

The DTC – Programming in Java – Level 1 course is targeted for beginners who want to:

- Learn how to think and write meaningful piece of code in Java.
- Learn how to read JAVA code that has been written by somebody else.
- Learn how to map literary description of a problem (requirement) to an application/library coded in Java. In summary, this course teaches how to program using Java programming language.

This is a core basic level course that is essential for anyone who have no prior programming experience but wish to be a professional Java engineer in future

## TARGET GROUP

- Anyone who has some basic knowledge about programming and wants to learn to write applications in JAVA for any purpose e.g. curiosity, hobby, to complete an academic project, to work towards a career as JAVA programmer, to help in project management, etc.

### Prerequisites:

- Basic knowledge about programming, bits/bytes, procedures, classes, computer architecture, etc. If you just have a theoretical knowledge that is perfectly okay but you should have strong convictions on what programming is, and what you hope to achieve from this class.
- Willing and eager to spend at least 10-20 hours (varying from student-to-student) per week outside of the training class to read/write codes in Java (self-study and practice).
- There is no prior educational level requirement for this course. Anyone from 10+2 student to someone who is doing her PHD in Genetic Engineering is welcome to take this course.
- If you are only interested in theory and have no interest/patience in spending at least 10 hours every week throughout the duration of the course, then this course is clearly not for you.
- If you have absolutely no idea about programming or do not see yourself doing programming in the next six -odd months, then this class may not be for you!

## TRAINING METHOD

- The course is spread over 40 hours that consists of lecture and lab work. There will be approximately 10 hours of lectures and 30 hours of hands-on lab work.
- Lab exercises are mandatory, have a fixed deadline, and are graded. The course puts heavy emphasis on lab exercises because software programming can only be learnt well by explicitly putting into practice the principles that have been taught (i.e. in simpler terms – by doing lots and lots of coding). Late submission (past the deadline) of exercises incur some penalty from total points.
- Instructors may provide relevant lecture/lab notes to students as (and when) necessary in the form of printed handouts and or via emails.
- Instructors may provide supplementary code snippets to students via email or in lab class to support the theory and or lab material that is being taught.
- At the end of the course, students may have to give an exam (which will be optional), that will test their knowledge on the material covered during the course. This exam may be practical and/or theoretical and is mandatory for any student wishing to join a higher level.
- Students are graded on the basis of attendance, lab exercises and exam in the increasing order of importance.

In summary, the only effective way to learn programming is to write lots of code. So, in order to really make this training productive, students are encouraged to spend as much time as necessary to complete the lab exercises on time. As part of the course, students will spend at least 30 hours in the lab but especially if you are new to programming or are coming from a non-computer-science background, it is recommended that you spend at least 10-20 hours per week outside of the class on your own to practice coding in Java.

## COURSE DURATION

- 40 hours
- Classes
  - ✓ Morning/Evening

## COURSE BREAKDOWN

### Theory:

1. OVERVIEW OF JAVA LANGUAGE
  - Introduction
  - H/w and s/w requirements
  - Installation of jdk.
2. PROGRAMMING WITH JAVA
  - Class declaration
  - Members of classes
  - Structure of java class
  - Main method
  - Command line arguments
  - Source code compilation
  - Coding convention
  - Java packages
3. CONSTANT, VARIABLES AND DATA TYPES
  - Primitives and non-primitives variables
4. DECISION AND BRANCHING
  - IF, ELSE, SWITCH, BREAK, CONTINUE
5. LOOPING
  - FOR, WHILE, DO-WHILE
6. FUNDAMENTALS OF LOOPS
  - Initializing objects
  - Static members
  - Inheritance
  - Polymorphism
  - Encapsulation
7. ABSTRACT CLASS AND INTERFACES
  - Defining interfaces
  - Separating interface and implementation
  - Implementing and extending interfaces
  - Abstract classes
8. EXCEPTION HANDLING
  - Exceptions and the exception hierarchy
  - Throwing exceptions
  - Catching exceptions
  - Chaining exceptions
  - The finally block
9. ADVANCE DATA STRUCTURES (JAVA COLLECTION CLASSES)
  - Arrays
  - List<e> interface and its implementation
  - Map<k,v> interface and implementation
  - Set<e> interface and implementation

## 10. JDBC CONNECTION

- Jdbc overview
- Using drivermanager, connection, statement, preparedstatement and resultset
- Create, delete, insert, update statements

## LABS

Lab assignments will focus on the practice and mastery of contents covered in the lectures; and introduce critical and fundamental problem-solving techniques to the students.

## DISCLAIMER

Please note that Deerwalk Training Center reserves the right to change the course syllabus of DTC – Programming in Java – Level 1 course at any time without prior notification.

# Programming in Java(Servlet, JSP & Spring Boot) : Level 2

## OBJECTIVES

The DTC – Programming in Java – Level 2 course is targeted for trainees:

- Who have had some prior beginner level hands-on programming experience in Java programming language.
- Who have programming experience in some other programming language (e.g. Java, Obj-C, PHP, C, C++, etc.) and want to learn Java.

## TARGET GROUP

- High school and university students (undergraduate, graduate, etc.) who want to do coursework (e.g. project, etc.) in JAVA.
- Someone who has experience in some other programming language (e.g. C/C++, PHP, Perl, etc.), but has never done programming in ANDROID.
- Someone who is already working as a professional VB.NET developer and wants to switch to ANDROID.
- Someone who did her undergraduate in Economics, has been working in Media sector since graduation, and also working as a professional freelance PHP developer.
- Electrical/Electronic undergraduates in their 3rd semester who want to beef up their software skills prior to graduation.

### Prerequisites:

- Successfully complete the entrance test with score of at least 40% (for trainees directly applying to this level).
- Successfully complete the DWIT Training – Programming in Java – Level 1 course (not applicable to trainees directly applying to this level).
- Successfully complete the interview.
- Willing and eager to spend at least 10-20 hours (varying from student-to-student) per week outside of the training class to read/write codes in Java (self-study and practice).

## TRAINING METHOD

The course is spread over 40 hours that consists of approximately 15 hours of lecture and 25 hours of hands-on lab work.

- Lab exercises are mandatory, have a fixed deadline, and are graded. The course puts heavy emphasis on lab exercises because software programming can only be learnt well by explicitly putting into practice the principles that have been taught (i.e. in simpler terms – by doing lots and lots of coding). Late submission (past the deadline) of exercises incur some penalty from total points.
- Instructors may provide relevant lecture/lab notes to students as (and when) necessary in the form of printed handouts and or via emails.
- Instructors may provide supplementary code snippets to students via email or in lab class to support the theory and or lab material that is being taught.
- At the end of the course, students may have to give an exam (which will be optional), that will test their knowledge on the material covered during the course. This exam may be practical and/or theoretical and is mandatory for any student wishing to join a higher level.
- Students are graded on the basis of attendance, lab exercises and exam in the increasing order of importance.

In summary, the only effective way to learn programming is to write lots of code. So in order to really make this training productive, students are encouraged to spend as much time as necessary to complete the lab exercises on time. As part of the course, students will spend at least 30 hours in the lab but especially if you are new to programming or are coming from a non-computer-science background, it is recommended that you spend at least 10-20 hours per week outside of the class on your own to practice coding in Java.

## COURSE DURATION

- 50 hours
- Classes
  - ✓ Morning/Evening

## COURSE BREAKDOWN

1. WEB APPLICATION BASICS
  - How the web works
  - Http overview, brief html review
  - Overview of java ee, servlets & web applications
2. SERVLET and JSP
  - Html forms
  - Http: request-response, headers, get, post
  - Overview: how servlets work
  - Requests and responses
  - Http servlets: httpServletRequest, HttpServletResponse and HttpServlet
  - Deployment descriptor
  - Accessing parameters
3. ADDITIONAL SERVLET CAPABILITIES
  - RequestDispatcher: including and forwarding
  - Sharing data with the request object attributes
4. USING CUSTOM TAGS
  - Custom tags to reduce jsp complexity
  - The jstl
  - Using custom tags
  - The c:url, c:param, c:foreach, c:out tags
5. SPRING BOOT
  - Technical requirements
  - Setting up the environment and tools
    - Installing IntelliJ
    - The basics of gradle and Maven
    - Creating the project with Spring Initializr
    - How to run the project
    - Spring Boot development tools
    - Logs and problem solving
    - Installing MariaDB and Mongo
6. USING JPA TO CREATE AND ACCESS A DATABASE.
  - Technical requirements
    - Basics of ORM, JPA, and Hibernate
    - Creating the entity classes
    - Creating CRUD repositories
    - Relationships between tables
    - Setting up the MariaDB database
7. CREATE A RESTFUL WEB SERVICE WITH SPRING BOOT.
  - Technical requirements
  - Creating a RESTful web service with Spring Boot



- Basics of REST
- Creating a RESTful web service
- Using Spring Data REST

#### 8. SECURING AND TESTING YOUR BACKEND

- Technical requirements
- Spring Security
- Securing your backend using JWT
- Testing in Spring Boot
- Creating unit tests

## LABS

Lab assignments will focus on the practice and mastery of contents covered in the lectures; and introduce critical and fundamental problem-solving techniques to the students.

## DISCLAIMER

Please note that Deerwalk Training Center reserves the right to change the course syllabus of DTC – Programming in Java – Level 2 course at any time without prior notification.

# Full-Stack Development in JAVA with Spring Boot and React : Level 3

## OBJECTIVES

This course builds on the foundation laid by DTC – Programming in Java – Level 3 to prepare trainees for a career as full-stack Java software engineer.

## TARGET GROUP

### Prerequisites:

- Successfully completed the DWIT Training – Programming in Java – Level 2 or obtained at least 40% score on the entrance exam.
- The latter case applies for new students that are directly attempting this training.
- Successfully complete the interview.
- Willing and eager to spend at least 10-20 hours (varying from student-to-student) per week outside of the training class to read/write codes in Java (self-study and practice).
- Please note that this is a lab intensive course where the students will be expected to work on lab exercises for approximately half the duration of the session.

## TRAINING METHOD

- The course is spread over 40 hours that consists of approximately 20 hours of lecture and 20 hours of lab work.
- Lab exercises are mandatory, have a fixed deadline, and are graded. The course puts heavy emphasis on lab exercises because software programming can only be learnt well by explicitly putting into practice the principles that have been taught (i.e. in simpler terms – by doing lots and lots of coding). Late submission (past the deadline) of exercises incur some penalty from total points.
- Instructors may provide relevant lecture/lab notes to students as (and when) necessary in the form of printed handouts and or via emails.
- Instructors may provide supplementary code snippets to students via email or in lab class to support the theory and or lab material that is being taught.
- At the end of the course, students may have to give an exam (which will be optional), that will test their knowledge on the material covered during the course. This exam may be practical and/or theoretical and is mandatory for any student wishing to join a higher level.
- Students are graded on the basis of attendance, lab exercises and exam in the increasing order of importance.

## COURSE DURATION

- 50 hours

- Classes
  - ✓ Morning/Evening

## COURSE BREAKDOWN

1. Setting Up the Environment and Tools – Frontend
  - Technical requirements
  - Installing Node.js
  - Installing VS Code
  - Creating and running a React app
  - Modifying a React app
2. Getting Started with React
  - Technical requirements
  - Basic React components
  - Basics of ES6
  - Understanding constants
  - Arrow functions
  - Template literals
  - Classes and inheritance
  - JSX and styling
  - Props and state
  - Component life cycle methods
  - Handling lists with React
  - Handling events with React
  - Handling forms with React
3. Consuming the REST API with React
  - Technical requirements
  - Using promises
  - Using the Fetch API
  - Practical examples
4. Useful Third-Party Components for React
  - Technical requirements
  - Using third-party React components
  - React Table
  - The modal window component
  - Material UI component library
  - Routing
5. Setting Up the Frontend for Our Spring Boot RESTful Web Service
  - Technical requirements
  - Mocking up the user interface
  - Preparing the Spring Boot backend.
  - Creating the React project for the frontend
6. Adding CRUD Functionalities
  - Technical requirements
  - Creating the list page
  - The delete functionality
  - The add functionality
  - The edit functionality

- Other functionalities
7. Styling the Frontend.
    - Technical requirements
    - Using the Button component
    - Using the Grid component
    - Using the TextField components
    - Using the AppBar component
    - Using the SnackBar component
  8. Deploying Your Application
    - Technical requirements
    - Deploying the backend
    - Deploying the frontend
    - Using Docker containers

## LABS

Lab assignments will focus on the practice and mastery of contents covered in the lectures; and introduce critical and fundamental problem-solving techniques to the students.

## DISCLAIMER

Please note that Deerwalk Training Center reserves the right to change the course syllabus of DTC – Programming in Java– Level 3 course at any time without prior notification.