



DIPLOMA IN JAVA PROGRAMMING

Nature of the Course: Theory + Practical

Total Hours per Day: 2 Hours

Course Duration: 2.5 Months + 1.5 Month (Internship)

Course Summary

This level 1 course is targeted for beginners who want to:

- Learn how to think and write meaningful pieces of code in java.
- Learn how to read JAVA code that has been written by somebody else.
- Learn how to map literacy description of a problem (requirement) to an application/library coded in java.

In summary, this course teaches how to program using the Java programming language. This is a basic level course that is essential for anyone who has no prior programming experience but wishes to be a professional Java engineer in future.

Completion Criteria

After fulfilling all of the following criteria, the student will be deemed to have finished the Module:

- Has attended 90% of all classes held.
- Has received an average grade of 80% on all assignments
- Has received an average of 60% in assessments.
- The tutor believes the student has grasped all of the concepts and is ready to go on to the next module.

Required Textbooks

- "Java Programming: From Problem Analysis to Program Design" by D.S. Malik
- "Java Software Solutions: Foundations of Program Design" by John Lewis, William Loftus, and Cara Cocking

Prerequisites

- Fundamental understanding of programming, bits/bytes, procedures, classes, and computer architecture. It's absolutely acceptable if you only have a theoretical understanding of programming, but you should be certain about what programming is and what you intend to gain for this session.

- If you are only interested in theory and have no interest/patience in spending at least 10hr every week throughout the duration of the course, then this course might not be for you.
- If you have absolutely no idea about programming or do not see yourself doing programming in the next six months, then this class may not be for you.

Course Details

Level 1

Overview Of Java Language

- Introduction
- Hardware and Software Requirement
- Installation of JDK

Programming With Java

- Class declaration
- Members of Classes
- Structure of Java Class
- Main Method
- Command Line Arguments
- Source code Compilation
- Coding Convention
- Java Packages

Constants, Variables And Data Types

- Primitive and non-primitive variables

Division And Branching

- If, else, switch, break, continue LOOPING
- For, while, do-while

Fundamentals Of Loops

- Initializing Objects
- Static Members
- Inheritance
- Polymorphism
- Encapsulation

Abstract Class And Interfaces

- Defining interfaces

- Separating interface and implementation
- Implementing and extending interfaces
- Abstract Classes

Exception Class

- Exception and the exception hierarchy
- Throwing exceptions
- Catching exceptions
- Chaining exceptions
- The ‘Finally’ block

Advance Data Structure (Java Collection Classes)

- Arrays
- List <e> interface and its implementation
- Map <k,v> interface and its implementation
- Set <e> interface and its implementation

Jdbc Connection

- JDBC overview
- Using driver manager, connection, statement , prepared statement and result set
- Create, delete, insert, update statements

PROGRAMMING IN JAVA (SERVLET, JSP & SPRING BOOT): LEVEL 2

Course Summary

The DTC-Programming in Java – level 2 course is targeted for trainees who have:

- Some prior beginner level hands-on programming experience in java programming language
- Programming experience in some other programming language (eg., Java, Obj-C, PHP, C, C++, etc.) and want to learn Java.

Completion Criteria

After fulfilling all of the following criteria, the student will be deemed to have finished the Module:

- Has attended 90% of all classes held.
- Has received an average grade of 80% on all assignments
- Has received an average of 60% in assessments.
- The tutor believes the student has grasped all of the concepts and is ready to go on to the next module.

Prerequisites

- Fundamental understanding of programming, bits/bytes, procedures, classes, and computer architecture. Its absolutely acceptable if you only have a theoretical understanding of programming, but you should be certain about what programming is and what you intend to gain for this session.
- If you are only interested in theory and have no interest/patience in spending at least 10hr every week throughout the duration of the course, then this course might not be for you.
- If you have absolutely no idea about programming or do not see yourself doing programming in the next six months, then this class may not be for you.

Course Details

Setting Up The Environment And Tools

- Technical requirements
- Installing NodeJS
- Installing VS Code
- Creating and running a React App
- Modifying a react App

Getting Started With React

- Technical requirement
- Basic react components
- Basics of ES6
- Understanding constants
- Arrow functions
- Template literals
- Classes and inheritance
- JSX and Styling
- Props and State
- Component lifeCycle Methods
- Handling Lists with React
- Handling Events with React
- Handling Forms with React

Consuming The Rest Api With React

- Technical requirement
- Using Promises
- Using the Fetch API
- Practical examples

Useful Third-Party Components For React

- Technical requirement
- Using third-party react components
- React Table
- The Modal Window Component
- Material UI Component Library
- Routing

Setting Up The Frontend For Spring Boot Restful Web Service

- Technical Requirements
- Mocking up the user interface
- Preparing the Spring Boot Backend
- Creating the React Project for the Frontend

Adding Crud Functionalities

- Creating a list page
- The Delete Functionality
- The Add Functionality
- The Edit Functionality
- Other Functionalities

Setting Styling The Frontend

- Technical requirement
- Using the bottom component
- Using the grid component
- Using the TextField Component
- Using the AppBar component
- Using the SnackBar component

Deploying Your Application

- Technical requirement
- Deploying the backend
- Deploying the frontend
- Using docker containers

Labs

Lab assignment will focus on the practice and mastery of content covered in the lecture; and introduce critical and fundamental problem-solving techniques to the students.

Learning Outcome

- Understanding the Java language syntax, rules, and core concepts.
- Developing foundational programming skills like problem-solving and logical reasoning.
- Learning object-oriented programming (OOP) principles, including encapsulation, inheritance, polymorphism, and abstraction.
- Gaining proficiency in writing and understanding Java code.