



קורס הכשרת מפתחי JAVA בהתמחות Spring



תיאור הקורס

מסלול ההכשרה המקיף להכשרת מפתחי JAVA בסביבת SPRING, המסלול יכשיר את הסטודנטים לעבודה מלאה כמפתחים עם מומחיות לצד שרת.

הסטודנטים יוכשרו לעבודה על כלים רבים ושונים שיאפשרו להם עבודה רציפה מקצה לקצה

מתודולוגית הלמידה:

הרצאות פרונטאליות, הדגמות, סימולציות בסביבת הלמידה ותרגול נרחב של לפחות 60%.
חומרי לימוד הכוללים דוגמאות והסברים "צעד אחר צעד".

היקף שעות הקורס

300 שעות לימוד אקדמאיות

דרישת קדם – רקע קודם בפיתוח תוכנה והיכרות עם תכנות ג'אווה

Module 1 – Java Programming

Computer structure and Algorithms	Computer structure. Low and high languages, Compiling, running and debugging
The components of JAVA language	language Types in JAVA, variables, assignment and conditional statements
Loop statements	The FOR statement, the WHILE statement, the DO-WHILE statement and casting.
Arrays	Arrays declaration ,multidimensional array
Methods in Java	Principles of procedural programming. Top-down design of a program, modulation, passing parameters to a method, static methods
Sorting and searching algorithms	Selection sort, insertion sort, bubble sort. Searching methods
Recursion	Recurrence as an alternative to iteration. Different kinds of recursion
Recursion	Recursion and arrays Towers of Hanoi problem
Object Oriented Basics	Class definition: attributes and methods, Constructors (copy, default), Setter and Getter methods, references to object, encapsulation.
Inheritance	Creating subclasses, overriding methods, class hierarchies.
Collections ,composite objects, self referential objects	Dynamic data structures: stack, linked lists, two way linked list

OOP Advanced	Polymorphism, Abstract Classes, Interfaces
Threads in Java	Java is a multi-threaded programming language which means we can develop multi-threaded program using Java.
Collections	The collections framework defines several algorithms that can be applied to collections and maps
Error Handling	<p>The Exception Handling in Java is one of the powerful mechanism to handle the runtime errors so that normal flow of the application can be maintained.</p> <p>In this page, we will learn about Java exceptions, its type and the difference between checked and unchecked exceptions</p>
Json+Rest	REST - Representational State Transfer) over JSON (JavaScript Object Notation) is the 1st choice when we try to expose services to the outside world. Though in recent times Web Sockets are making developers think as a replacement as REST APIS are too chatty and increase the messages sent over the network
Developing Server in Java	Server Development using Java. We will learn how to build our first Java Server.
DB Connection	We will learn how to connect Java Server to DB using SQL Connection and how to build simple 2 Tears Server – DB Java application

Module 2 – Spring

- Introduction to Spring
 - Setting Up Your Development Environment
 - Spring Framework Need and Introduction
 - Spring Building Blocks

BlueTech

we create new talents • Spring Events and Tools



- What is Spring framework?
- Why Spring framework?
- Spring framework architecture
- Usage scenario
- Tools (we are going to use in this course)
- Spring Helloworld Refactoring
- Spring Inversion of Control - XML Configuration
- Spring Bean Scopes and Lifecycle
- Spring Configuration with Java Annotations - Bean Scopes and Lifecycle Methods
- Spring Dependency Injection Basics and Advanced
 - What is and Why Dependency Injection (DI)?
 - Dependency Injection variants
 - DI classes in Spring framework
 - DI parameter types
 - XML Configuration
 - Bean naming
 - @Autowired annotation
 - Auto-scanning
- Spring Dependency Injection Annotation
 - Annotation-based Dependency Injection
 - Qualifier
 - JSR 330 (Dependency Injection for Java)
 - JSR 250 (Common Annotations)
 - Java-based Spring configuration (instead of XML configuration file)
 - @Component and further stereotyped annotations
 - Auto scanning
- Spring 5 and Maven
 - Maven Repositories for Spring artifacts



- Creating Spring Maven projects

- Spring 5 the gradle

- Introduction to Gradle

- Basic of Gradle

- Repositories and Artifacts

- Spring MVC

- Spring 5 MVC Basics I

- Introduction to Spring MVC

- Controller

- Annotation-based controllers

- Request mapping

- URI template

- Request mapping with other means

- Handler method arguments

- Handler methods return types

- Spring 5 MVC Basics II

- Handler mapping customization

- Resolving views

- XML namespaces brief tutorial

- Spring 5 <mvc:...> configuration

- Spring 5 MVC Form handling

- 2-phase form submission handling

- Command/form objects

- @ModelAttribute

- Data binding

- Validation

- Redirect

- Form tags

- Spring 5 MVC Misc.

- Exception handling

- Locale handling

- @Value

- SpEL

- Spring 5 MVC Views

- ContentNegotiatingViewResolver

- Tiles

- Velocity and Freemarker

- PDF or Excel

- JasperReports

- Spring 5 MVC REST

- Annotation-based REST support

- URI template

- Content negotiation

- Feed views

- XML marshaling views

- JSON mapping views

- HTTP method conversion

- RestTemplate

- Exception handling

- Message conversion

- Spring 3 MVC Ajax

- Spring 5 AOP

- What is and Why AOP?

- AOP concepts and terminology

- @AspectJ support in Spring

- Types of advice

- Declaring advices

- Accessing Join point information

- Declaring a pointcut

- Defining and using common pointcuts

- Introduction

- Spring 5 Database
 - @Repository annotation
 - MongoDB
- Data access through JDBC
 - JdbcTemplate class
 - NamedParameterJdbcTemplate class
 - SimpleJdbcTemplate and SimpleJdbcDaoSupport classes
- Data access through ORM
 - Hibernate
 - JPA
- Spring 5 Transaction
 - Transaction management in Spring framework
 - Global transaction vs. local transaction
 - PlatformTransactionManager interface
- Declarative transaction management
 - XML-based
 - Annotation-based
 - Transaction propagation
- Spring 5 Testing
 - Unit testing without using Spring DI \ JUnit
 - Unit testing using Spring DI
 - Spring TestContext framework
 - Transaction-aware testing
 - Spring MVC Testing
 - Spring Testing Annotations
- Spring – Springboot
 - Intro
 - Developing a REST API Controller with Spring Boot
 - Spring Boot - Spring Boot Dev Tools and Spring Boot Actuator

- Spring Boot - Running Spring Boot Apps from the Command Line
- Spring Boot - Application Properties
- Spring Boot - Build a REST CRUD API with Hibernate
- Spring Boot - Spring Data REST - Real-Time Project
- Spring 5 - Security
 - Spring Security - Adding Custom Login Form
 - Spring Security - Bootstrap CSS Login Form
 - Spring Security - Adding Logout Support
 - Spring Security - User Roles and Restrict Access Based on Role
 - Spring Security - Add JDBC Database Authentication
 - Spring REST - Overview

Module 3 - Learn Swagger and the Open API

- Introduction and tools
- YAML and YAML vs JSON
- SwaggerHub - An Overview
- Open API Specification
 - Step 1: Add the OpenAPI Version and Meta data
 - Adding: Server Details , Paths , Tags to Group API Operations
 - Data Types in OpenAPI
- Swagger vs OpenAPI

Module 4 - API development

- Introduction to RESTful API
- Private, Public and Partner API
- REST API Architectural Constraints
- REST API Design and concerns
- REST API Error Handling Patterns
- REST API Handling Change - Versioning Patterns
- REST API Cache Control Patterns

- REST API Response Data Handling Patterns



- REST API Specifications using Swagger
- API Management
- API testing using Postman

Module 5 - MongoDB

MongoDB is the *hottest* database solution in the tech world right now, able to power anything from your growing personal project to an enterprise infrastructure.

- Windows Setup
- Core Fundamentals of MongoDB
- Mongo Operators
- Handling Relational Data
- Schema Design

Jenkins

- Jenkins CI Server - overview
- Basic Jenkins system configuration
- Jenkins plugins
- Build Nodes - Windows & Linux
- Jenkins build jobs:
 - Freestyle Project & Maven Project
 - Configuring SCM
 - Build steps
 - Post-build steps
 - Adding parameters
 - Sending Mail
- Continuous Delivery Pipelines:
 - Job dependencies and triggering
 - Build Pipeline Plugin
 - Delivery Pipeline Plugin
- Monitoring and Tuning Jenkins

Docker:

Docker is an open platform for developers and sysadmins to build, ship, and run distributed applications, whether on laptops, data center VMs, or the cloud.



- Introduction
- Docker Commands
- Docker Run \ Images \ Compose \ Registry
- Docker Engine, Storage and Networking
- Container Orchestration - Docker Swarm & Kubernetes

BlueTech

Kubernetes:

we create new talents

Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications. It was originally designed by Google and is now maintained by the Cloud Native Computing Foundation.

Kubernetes is at the cutting-edge of application deployment. The best way to kick-start your DevOps career is by learning how to effectively deploy Kubernetes.

- Kubernetes Overview
- Setup Kubernetes
- Kubernetes Concepts
- Services and Microservices Architecture

OPEN SHIFT OCP

OpenShift® Container Platform is a consistent hybrid cloud foundation for building and scaling containerized applications

- Intro
- Getting started
- Concepts - Builds and Deployments
- Networks, Services, Routes and Scaling
- Storage, Templates and Catalog

Jira

- Introduction
- Agile Concepts Refresher & Jira Terms
- Working within an Agile Team - For general team members
 - Intro to the JIRA user interface & navigation
 - Intro to Project Navigation
 - Intro to Issue Navigation
 - Viewing, editing & understanding Issues
 - Working with Agile Boards
 - Agile Board - Backlog View
 - Creating Issues
 - Searching for Issues
 - Advanced searching using JQL
 - Search filters

BlueTech

we create new talents

o Dashboard customization

o Understanding the Resolution Field



Jfrog

- What is a Artifactory?
- What is a repository?
- Introduction – Why to use Jfrog
- Create and build a simple project

