Full Stack Development

JAVA FULL STACK DEVELOPER TRAINING

500K+	100K+	300+	300K+	200+	150+
Satisfied Students	Online Students	Trainers	Placements	Global Certifications	Companies

www.cromacampus.com

UI Development

Introduction

- UI Developer roles and Responsibilities
- UX designer roles
- Technologies needed
- Power of UI
- Current market requirements on UI
- Sample Webpages
- Crawling and meta tags

Basics

- DOM
- Structure of HTML Page
- Mandatory tags in html page (html, head, body)
- What is CSS
- Different ways of applying css for elements, and priority chain of css
- Heading tags(H1...H6), Tags and attributes(Class, Id, style..etc)
- Inline and blocklevel elements

More Tags In Html

- Including external page links in a page using anchor tags and its
- properties
- Working with row and column data using table tags
- Hiding and unhiding elements using display property
- img tag, p tag, ul and ol tags, li, nobr, hr, br etc
- Layouts, forms, buttons
- Input fields (textbox, radiobutton, checkbox, dropdown, textareaetc)

HTML5

- Difference between HTML5 and HTML 4
- List of Browsers support HTML5
- Doctype
- Media tags (audio and video tags)
- Graphics using Canvas tag
- Drag and Drop features
- Working on locations lat and lng using Geolocation
- Storing userpreferences using Localstorage.

More CSS Properties

- Adding borders, font, pseudo classes
- 00 | www.cromacampus.com
- Positioning elements (absolute, relative, fixed and static)
- Image spriting
- Boxmodel (margins, padding)
- Floating elements (float left, right etc.)
- Including external resources

- Absolute and Relative paths
- Including external resources like css, images etc

Form Elements

- Get & Post
- Validating input values in a form
- Form action and type

CSS 3

- Difference between CSS2 and CSS3
- Adding borders and backgrounds
- Advanced text effects(shadow)
- 2D and 3D Transformations
- Adding Transitions to elements
- Adding animations to text and elements

JavaScript

- Introduction
- Data types and data structures in Js
- Control structures, if, ifelse, while, for, switch case statements
- Dynamic creation and maniplation of dom elements using js
- Adding dynamic event listerners to dom elements
- Event capturing and event bubbling
- Validations using key charcodes

JavaScript Supported Data Structures

- Arrays
- Predefined methods in arrays
- Strings and predefined methods
- Objects
- JSON

Advanced JavaScript

- Prototyping in JavaScript
- Closures in JavaScript
- Inheritance in JavaScript
- Adding methods for an object

Bootstrap 5

- Get Started?
- What is Bootstrap?
- Bootstrap History
- Why Use Bootstrap?
- What Does Bootstrap Include?
- HTML File
- Adding Bootstrap to Your Web Pages
- Downloading Bootstrap
- Bootstrap CDN

- Put Your HTML Elements Inside Containers
- Typography
- Colors
- Tables
- Images
- Jumbotron
- Alerts
- Buttons
- Button Group
- Button Dropdown
- Badges
- Progressbars
- Pagination
- List Groups
- Cards
- Collapse
- Navbar
- Forms
- Inputs
- Carousel
- Modal
- Tooltip
- Popover
- Scrollspy
- Utilities
- Grid Sytem
- BOOTSTRAP RESPONSIVE LAYOUT PROJECT

JQuery Framework

- Onload and onreadydiffrence
- jQuery selectors
- Multiple ways of referingdom elements using jQuery selectors
- jQuery methods
- Adding dynamic properties for dom elements
- Toggleing elements
- Creating dynamic elements using jQuery

JQuery Traversing Methods

- Finding elements using jQuery techniques
- Filtering elements

Events Using JQuery

- Binding events
- Dynamic binding
- List of events been supported in jQuery(blur, change, click, dbclick....etc)

AJAX

- Advantages with Ajax and its limitations

- Samples working with Ajax
- Different data formats used in Ajax (string, xml, Json, etc)
- XML and JSON difference
- Crossdomain interactions using JSONP

JQuery Templating

- Loading DOM dynamically using jQuery templates
- loading templates using AJAX

React-JS

- Foundation to Reactjs
- Introduction to Reactjs
- Introducton to concepts on ES6 (ECMASCRIPT)
- let and const
- Arrow functions
- Template literals
- Array method (map)
- Array & Object destruction
- Spread Operator
- classes
- constructor
- this usage in class
- getters & setters in class

React Basics

- introduction to webpack & babel introduction.
- creating project with ReactJs, using npm & npx.
- ReactJs folder structure.
- creating "hello world" inital programe.
- using commands to run & build project.
- introduction to Jsx
- inlcude dynamic javascript expression in Jsx
- attributes in Jsx
- using dot notation in Jsx
- best practices in crating Jsx components
- Jsx childrens & group elements
- rendering & updating elements on ReactJs
- introduction to functional components & working with it.
- introduction to class components & working with it.
- differenc between class & functional components
- ReactJs components and props
- styling with css in ReactJs
- ReactJs state & setState
- lifecycle methods in ReactJs

ComponentDidMount()

ComponentDidUpdate()

- ShouldComponentUpdate()
- ComponentWillUnmount()
- Handling Events In ReactJs
- Conditional Rendering In ReactJs
- Forms In ReactJs

Advanced ReactJs

- Using Fetch & Promises & Async Await In ReactJs To Get API Data.
- Fragments In ReactJs
- Pure Components In ReactJs
- Memo In Functional Components
- Refs In ReactJs
- Refs In Class Components
- Portals In ReactJs
- Higher Order Functions
- React Router
- In Details Of Switch, Router, Path, Link, UseParams, UseRouteMatch
- Router Paramters
- Router Nesting
- Transitions With Router
- Router Config
- Router Redirecting
- Developing Router Practical Application
- Redux In React
- Creating Store With Redux
- Data Flow In Redux

- Redux Actions & Reduces
- Developing Redux Practical Application
- Advanced Context API
- React.CreateContext
- Context.Provider
- Class.ContextType
- Context.Consumer
- Context.DisplayName
- Fetching Data With Use Effects
- Static Type Checking
- Error Boundaries
- Typechecking With PropTypes
- Introducing Hooks In ReactJs
- Use State In Hooks
- Use Effects In Hooks
- REACT JS Content Will Be Updated As Per Latest Updations

Core Java

- Introduction:
  - Java History
  - Differences between java and others
  - Java Features
  - Java Naming Conventions
- First Java Application Development:
  - Java Installation
  - Editor
  - Java Application and Java File Saving.
  - Compile Java File
  - Execute Java Applications.
- Language Fundamentals
  - Operators

- Identifiers
- Literals
- Data Types and Type casting
- Java Statements
- Arrays

OOPS

- Introduction
- Class
- Object
- Encapsulation
- Abstraction
- Inheritance
- Abstraction
- Polymorphism
- Message Passing
- Object Based PL VS Object Oriented PL
- Class syntax
- Method Syntax
- Var-arg method.
- Accessor Methods VS Mutator Methods
- Syntax to create an object
- Immutable Objects VS Mutable Objects
- Object Vs Instance
- Constructors
- Default Con.
- User defined con.
- 0-arg-con.
- param-con.
- Instance Context
- Instance variable
- Instance block.
- Instance method
- This keywords
- To refer current class variable.
- To refer current class methods.
- To refer current class blocks.
- To return current class objects.
- Static keyword
- Static variable
- Static block
- Static method
- Static import
- Main () method
- Public static void main (String [] args)
- Why public?
- Why static?

- Why void?
- Why main?
- Why String [] as parameter?
- Is it possible to overload main (-) method?
- Is it possible to override main (--) method?
- Is it possible to provide more than one main (--) method with in a
- single java appl?
- Is it possible to execute any java application without using main
- method?
- Factory Method
- Singleton classes and Doubleton classes
- Final Keyword
- Final variable
- Final class
- Final method
- Enum keyword
- Relationships in JAVA
- IS-A Vs HAS-A Vs USE-A
- Associations in Java
- one-one
- many-one
- one-many
- many-many
- Inheritance and Types of inheritances
- Single
- Hierarchical
- Multiple
- Hybrid.
- Multilevel
- Static flow in inheritance
- Instance flow in inheritance
- Super keyword
- Class level type casting
- Poly Morphism
- Static PM
- Dynamic PM
- Method overloading
- Method overriding
- Abstract Methods Vs Concreate Methods
- Abstract class Vs concrete Class
- Class Vs Abstract class Vs interface
- Instance of operator
- JAVA8 features in interfaces
- What is Adapter class?
- What is marker interface?
- Object Cloning
- Shallow Cloning

- Deep Cloning

Inner Classes:

- Member Inner class
- Method local Inner class
- Static Inner class
- Anonymous Inner class

Wrapper Classes:

- Byte
- Short
- Integer
- Long
- Float
- Double
- Boolean
- Character

Packages:

- What is a package?
- Adv. of packages
- Modularity
- Reusability
- Abstraction
- Sharability
- Security
- Types of packages
- Predefined packages
- User defined packages
- Jar files preparation
- Executable Jar files
- Batch files preparation

String Manipulations:

- String
- String Buffer
- String Builder
- String to kenizer

Exception Handling:

- Error VS Exception
- Exception Def.
- Types of Exceptions
- Predefined Exceptions
- User defined Exceptions
- Checked Exception VS Unchecked Exception
- Pure Checked Exceptions
- Partially Checked Exceptions

- Throw Vs throws
- try-catch-finally
- Custom Exceptions
- Java7 Features in Exception Handling
- Automatic Resource management
- Multi catch block.

Multi-Threading:

- Process Vs Processor Vs Procedure
- Single Processing Mech. Vs Multi Processing Mech.
- Single Thread model And Multi Thread Model
- Thread Design
- Extending Thread class
- Implementing Runnable interface.
- Thread lifecycle
- New/Born
- Runnable
- Running
- Blocked
- Dead
- Thread class library
- Sleep ()
- Join ()
- Yield ()
- Stop ()
- Thread class library
- Synchronization
- Inter Thread communication
- Wait ()
- String to kenizer
- Notify ()

IOStreams:

- What is stream?
- Types of Streams?
- Byte-oriented Stream
- Input Streams
- Output Streams
- Character-Oriented Streams
- Reader
- Writer
- File Input Stream Vs File Output Stream
- File Reader Vs File Writer
- File Vs Random Access File
- Serialization vs Deserialization
- Externalization

Networking:

- Standalone Appl. Vs Distributed Appl.
- Client-Server Arch.
- Socket Vs Server Socket
- Network Appl. Arch.
- Socket Programming.

Reflection API:

- Class
- Field
- Method
- Constructor

Annotations:

- What is Annotation?
- Adv of annotations
- Comments Vs Annotations
- Types Of annotations
- Built-in Annotations
- Override
- Inherited
- Deprecated
- Target
- Suppress Warnings
- Documented
- Retention
- User Defined Annotations

Regular Expressions:

- Introduction
- Pattern
- Character
- Quantifiers

Garbage Collection:

- Introduction
- Approaches to make an object for GC
- Methods for requesting JVM to run GC
- Finalization

JVM Arch.

- Class Loading Sub System
- Memory Management System
- Execution Engine
- Java Native Interface
- Java Native library

Generics:

- Introduction

- Generic Classes
- Generic Methods & Wild Card Character.
- Inter Communication with Non-Generic Code

ADV. JAVA

Collection Framework:

- Collection Arch.
- List and its implementations
- Array List
- vector
- LinkedList
- stack
- Set and its implementations
- Hash Set
- Linked Hash Set
- Tree Set
- Map and its implementations
- Hash Map
- Hash table
- Properties
- TreeSet
- Queue and its implementations
- Priority Queue
- Blocking Queue
- Priority Blocking Queue
- Linked Blocking Queue
- Iterators
- Iterator
- List Iterator
- Enumeration
- Message Passing

JDBC:

- Storage Areas
- Temporary Storage Area
- Permanent Storage Areas
- Query Processing System
- Query Tokenization
- Query Processing
- Query Optimization
- Query Execution
- Driver and Driver Types
- Type 1 Driver
- Type 2 Driver
- Type 3 Driver
- Type 4 Driver

- Steps to design JDBC Applications
- Load and register the Driver.
- Establish the connection between Java Application.
- Prepare either Statement or prepared Statement or Callable
- Statement Objects.
- Write and execute SQL Queries.
- Close the connection.
- Prepared Statement
- PreparedStatement with insert sql query
- PreparedStatement with update sql query
- PreparedStatement with select sql query
- PreparedStatement with Dates Handling
- PreparedStatement with Batch Updations
- Callable Statement
- CallableStatement with procedure
- CallableStatement with function
- CallableStatement with CURSOR Type Procedure
- CallableStatement with CURSOR type function
- Transaction Management
- Atomicity
- Consistency
- Isolation
- Durability
- Savepoint
- Batch Updations

SERVLETS:

- Introduction
- Enterprise Applications
- Web Applications
- Distributed Applications
- Client-Server Arch
- Client
- Protocol
- Server
- Servlets Design
- Servlet interface
- Generic Servlet
- Http Servlet
- Servlet Lifecycle
- Servlet Config
- Servlet Context
- Servlet Communication
- Browser-servlet
- SendRedirect Mechanism
- Web-component
- Include Mechanism

- Forward mechanism
- Session Tracking Mechanisms
- HttpSession Session Tracking Mechanism
- Coockies Session Tracking Mechanism
- URL-Rewriting Session Tracking Mechanism
- Hidden Form Fields Session Tracking Mechanism
- Servlets Filters

JAVA SERVER PAGES:

- Introduction
- JSP Life Cycle
- JSP Elements
- JSP Directives
- Scripting Elements
- JSP Actions
- JSP Directives
- Page Directive
- Include Directive
- Taglib Directive
- JSP Scripting Elements
- Declarations
- Scriptlets
- Expressions
- JSP implicit objects
- Out
- Request
- Response
- Config
- Application
- Session
- Exception
- Page
- Page Context
- JSP Standard Actions
- <jsp:useBean>
- <jsp:setProperty>
- <jsp:getProperty>
- <jsp:include>
- <jsp:forward>
- <jsp:param>
- JSP Custom Actions
- Tag
- IterationTag
- BodyTags
- JSTL
- Core Tags
- XML Tags

- Expression Language
- EL operators
- EL implicit objects.
- EL functions.

Hibernate Framework

Introduction

- Enterprise
- Enterprise Application
- Enterprise Application Layer
- User Interface Layer
- Business Processing Layer
- Data Storage and Access Layer
- Data Persistency
- Data Persistency through Serialization and Deserialization
- Data Persistency through JDBC
- Data Persistency through ORM
- Paradigm Mismatches
- Granualarity Mismatch
- Sub Types Mismatch
- Associations Mismatch
- Identity Mismatch
- EJBs Vs Hibernate
- JPA Vs Hibernate
- Hibernate History
- Hibernate Features
- Hibernate Arch.

Steps To Prepare Hibernate Application

- Persistence Class / POJO class
- Mapping File
- Hibernate Configuration File
- Client Application

Hibernate Applications

- Hibernate Application with Main Class as Client.
- Hibernate Application with GUI Application as Client.
- Hibernate Application with Servlet as Client.
- Hibernate Application with JSP Page as Client.
- Hibernate Application with Struts Application as Client.
- Hibernate Application with MYSQL DB
- Hibernate Application with Multiple DBs [Oracle DB and MySQL DB]
- Hibernate Basic Annotations [Without Mapping File]
- Hibernate Application without Configuration File
- Hibernate Application with Composite Keys.

Hibernate Persistence Object Lifecycle

- Transient State
- Persistent State
- Detached State
- Removed State

Hibernate Tools

- Schema Export
- Schema Update
- Code Generation

Primary Key Generation Algorithms [XMl And Annotations]

- Assign
- Increment
- Sequence
- Identity
- Hilo
- Seq-Hilo
- Native
- UUID
- Foreign
- GUID
- Select

Transaction Management

- ACID Properties
- Automicity
- Consistnacy
- Isolation
- Durability
- Transaction Management in JDBC
- Automicity Achievement in JDBC
- Isolation Problems
- Transaction Management in Hibernate

Hibernate Query Language [HQL]

- HQL Elements
- Clauses
- 'From' Clause
- 'Select' Clause
- 'Where' Clause
- 'Order by' Clause
- 'Group by' Clause
- 'Having' Clause
- Aggregate Functions
- count(-)
- sum(-)
- min(-)
- max(-)

- avg(-)
- Generic Expressions
- Arithmetic Operators in Generic Expressions
- Comparision Operations in Generic Expressions
- Scalar Functions in Generic Expressions
- 'In
- 'Between
- 'Like
- 'is null
- 'is not null
- Parameters
- Positional parameters
- Names Parameters
- Subqueries
- Pagination
- HQL with Updations

Native SQL

- Scalar SQL Queries
- Stored Procedures and Functions

Criteria API

Hibernate Filters

Hibernate Mappings

- Basic 'OR' Mapping
- Component Mapping
- Inheritance Mapping
- Table per Class Hierarcy
- Table per Sub-Class
- Table per Concreate Class
- Associations Mapping
- One-To-One Association
- One-To-Many Association
- Many-To-One Association
- Many-To-Many Association

Connection Pooling

- Inbuilt Connection Pooling Support in Hibernate.
- Third Party Connection Pooling Mechanisms C3P0, Proxool, DBCP.....
- Connection Pooling through Weblogic Server JNDI.

Cache Mechanisms

- I level Cache
- II Level Cache

Spring

Introduction

- Enterprise Appl
- Enterprise Application Layers
- Presentation Layer
- Business Layer
- Data Access Layer
- System Architectures
- 1-Tier Arch.
- 2-Tier Arch.
- n-Tier Arch
- Types of Enterprise Applications.
- Web Applications
- Distributed Applications
- Modeled Arch.
- Model-I Arch.
- Model-II Arch.
- MVC
- Requirement to user Frameworks
- Types of Frameworks
- Web Frameworks
- Application Frameworks
- Differences between Spring and Struts, JSF
- Spring History
- Spring Modules.
- Spring1.x Modules
- Spring2.x Modules
- Spring3.x Modules
- Spring4.x Modules
- Spring5.x Modules

Steps To Prepare Spring Application

Core Module Application:

- Download Spring Framework from Internet.
- Provide Spring Setup in Eclipse IDE
- Prepare Bean Class
- Prepare Bean Configuration File
- Prepare Test / Client Appl.

Core Module

- Introduction
- IOC Containers
- BeanFactory
- XmlBeanFactory
- Resources
- ByteArrayResource
- FileSystemResource

- ClassPathResource
- InputStreamResource
- UrlResource
- ServletContextResource
- PortletContextResource
- ApplicationContext
- ClassPathXmlApplicationContext
- FileSystemXmlApplicationContext
- WebXmlApplicationContext
- Beans in Spring Framework
- Beans Definition
- Beans Configuration
- XML Based Configuration
- Annotation Based Configuration
- Java Based Configuration
- Bean Scopes
- singleton Scope
- prototype Scope
- request Scope
- session Scope
- globalSession Scope
- application Scope
- webSocket scope
- Custom Scopes in
- Spring Framework.
- Bean Lifecycle
- Bean Loading
- Bean Instantiation
- By Constructor
- By Static Factory Method
- By Instance Factory Method
- Bean Initialization and Destruction
- By Custom initialization and destruction methods.
- By InitializingBean and DesposableBean callback interfaces.
- By @PostConstruct and @Predestroy Annotations
- Beans Inheritance
- Nested Beans
- BeanPostProcessor
- Inversion Of Control[IOC]
- Dependency Lookup
- Dependency Pull
- Contextualized Dependency Lookup
- Dependency Injection
- Constructor Dependency Injection
- Setter Method Dependency Injection
- Different Types of Elements Injection
- User defined data types elements injection.

- List types injection
- Set types injection
- Map Types Injection
- Properties types Injection
- Circular Dependency Injection
- Name Spaces
- P-Name space
- C-Name Space
- Beans Autowiring or Beans Collaboration
- Autowiring and its Modes
- no
- byName
- byType
- constructor
- Annotation Based Wiring
- Autodiscovery or Stereo Types
- Java based Autowiring[Java Based Configuration]
- Method Injection
- Lookup Method Injection
- Arbitrary Method Replacement
- Event Handling
- ContextRefreshedEvent
- ContextStartedEvent
- ContextStoppedEvent
- ContextClosedEvent
- RequestHandledEvent
- Custom Events In Spring Framework
- Bean Validations in Spring Framework
- Internationalization in Spring Framework
- Bean Manipulations and Bean Wrappers
- Property Editors
- ByteArrayPropertyEditor
- ClassEditor
- CustomBooleanEditor
- CustomCollectionEditor
- CustomNumberEditor
- FileEditor
- InputStreamEditor
- LocaleEditor
- PatternEditor
- PropertiesEditor
- StringTrimmerEditor
- URLEditor
- Custom Property Editors
- [USer defined]
- Profiling
- Spring Expression Language[SpEL]

- SpEL Expressions
- SpEL Operators
- SpEL Variables
- SpEL Medthod Invocations
- SpEL Collections

Spring JDBC/DAO Module:

- Introduction
- DAO Definition
- Advantages of DAOs
- Drawbacks with DAOs
- Guidelines to prepare DAOs
- Pain JDBC Vs Spring JDBC
- JdbcTemplate
- NamedParameterJdbcTemplate
- Parameter values through Map
- Parameter Values through SqlParameterSource
- MapSqlParameterSource
- BeanPropertySqlParameterSource
- SimpleJdbcTemplate
- DAO Support Classes
- JdbcDaoSupport
- NamedParameterJdbcDaoSupport
- SimpleJdbcDaoSupport
- Spring Batch Updations or Batch Processing
- Stored Procedure and Functions in Spring JDBC
- Procedures and Functions without CURSOR Types
- Procedures and Functions with CURSOR Types
- Blob and Clob processing in Spring JDBC
- AbstractLobCreatingPreparedStatementCallback
- AbstractLobStreamingResultSetExtractor
- LobCreator
- LobHolder
- Connection Pooling in Spring JDBC
- Default Connection Pooling Mech.
- Third Party Connection Pooling Mechanisms
- Apache DBCP
- C3P0
- Proxool
- Application Servers provided Connection Pooling Mechanism
- Weblogic12c provided Connection Pooling Mechanism.

Spring ORM

- Introduction
- Hibernate Integration with Spring
- Hibernate Introduction
- Hibernate Application Development

- Spring with Hibernate Integration.
- JPA Integration with Spring
- JPA Introduction.
- JPA Application development
- Spring with JPA Integration.

Aspect Oriented Programming [AOP]

Introduction

AOP Terminalogy

- Aspect
- Advice
- JoinPoint
- Pointcut
- Introduction
- Target
- Proxy
- Weaving
- Advisor
- Types of AOPs
- Proxy Based AOP
- Declarative Based AOP
- Annotation Based AOP
- Advices
- Before Advice
- After Advice
- After-Returning Advice
- Around Advice
- After-Throwing Advice
- Pointcuts
- Static Pointcut
- Dynamic Pointcut.

Spring Transactions

- Introduction
- Transaction Attributes
- Isolation Levels
- Programmatic Based Transactions
- Declarative Based Transactions.
- Annotation Based Transactions

Spring Web MVC Module

- Introduction
- Spring MVC Flow
- Controllers
- Abstract Controller
- ParameterizableViewController

- MultiActionController
- Command Controllers
- AbstractCommandController
- AbstractFormController
- SimpleFormController
- AbstractWizardFormController
- Handler Mappings
- BeanNameUrlHandlerMapping
- SimpleUrlHandlerMapping
- HandlerInterceptor
- ViewResolvers
- AbstractCachingViewResolver
- XmlViewResolver
- ResourceBundleViewResolver
- UrlBasedViewResolver
- InternalResourceViewResolver
- VelocityViewResolver / FreeMarkerViewResolver
- Spring Exception Handling
- File Uploading and File Downloading
- Internationalization
- Spring MVC with Tiles

Spring Web

- Introduction
- Spring Integration with Struts.
- Spring Integration with JSF.

Spring Security

- Spring Security Introduction
- Spring Security Features
- Spring Security XML Based Example
- Spring Security Java Based Example

SpringBoot and Microservices

Spring Overview

- What is Spring Framework?
- What is Framework?
- What is Enterprise Edition?
- EE Vs Spring
- Spring5 Architecture
- Spring Projects Overview

Spring Core Terminology

- Tight Coupling Vs Loose Coupling(Factory Design Pattern)
- Inversion Of Controller(Design Principle)
- Dependency Injection(Design Pattern)
- IOC Container

- Beans
- Autowiring
- Bean Factory
- Application Context
- SpEL(Spring Expression Language)

Spring Core Basics

- Creating a Project
- Dependency
- Decoupling Components
- Managing Beans and Dependencies
- Contexts and Dependency Injection Framework
- Spring Application Configuration
- XML Application Configuration
- XML Configuration with Java Annotations
- Stereotype Annotations
- Using an External Property File
- Autowiring By Type — @Primary
- Autowiring By Name
- Autowiring — @Qualifier
- Constructor and Setter Injection

Spring Core In-Depth

- Bean Scope
- Mixing Bean Scope
- @ComponentScan
- Bean Lifecycle:
- @PostConstruct
- @ PreDestroy
- Prototype Scoped Beans

Spring Boot

- Overview
- Features Of Spring Boot
- Creating a Spring Boot Project Using
- Spring Boot Command Line Interface(CLI)
- Spring Initializer(https://start.spring.io/)
- Spring Tool Suite(STS)

Spring Boot Application Components

- POM file
- Spring Boot Starters
- Application.java
- application.properties
- ApplicationTests.java

Spring Boot Application Packaging Internals Spring

Boot Auto Configuration

- Convention Over Configuration/Coding(C2C)
- Understanding the Spring boot Auto Configuration
- Enabling/Disabling the Spring Boot auto-configuration

Alternating The Spring Boot Configurations

- Overriding the default Configuration Values
- Externalizing the properties from Spring Boot Application
- Creating the Custom Configuration Properties
- Reading the Custom Configuration Properties
- Changing the Default Embedded Server
- Deploying the application in application servers I.e. Tomcat and Jboss.

Understanding The Connection Pooling

- Hikari CP Connection Pooling

Profiles

- What is Profile?
- Maven profiles in traditional web applications
- Conditionally creating beans with profiles
- Spring Boot Profiles Using
- Multiple properties file
- yml file

Implementing Data Access Layer

- Spring Data JPA
- Hibernate

Implementing Service Layer

- Transactions Support in Spring Boot
- Local Vs. Global Transactions
- Understanding the TransactionStatus I.e. rollback
- Understanding the TransactionDefinition I.e.

ISOLATION Level Attributes

Propagation Behaviour Attributes

Timeout

Implementing Controller Layer

- Building Spring Rest Controller Components using Spring Boot
- API Versioning and Content-Negotion
- Building Spring web mvc Controller Components using Spring Boot
- Validatons and Databindings
- Internationalization

Testing Spring Boot Applications

- Testing Web MVC Controllers
- Testing Spring Rest Controllers
- Unit Testing with Mockito

Spring Boot Logging

- Understanding the Spring Boot Default logging
- Configuring the logs using: slf4j and logback

Actuators

- Health Check
- Metrics
- Environment .etc
- Spring Boot Admin
- Micrometer

Overview Of Spring Boot Security

Microservices

- Domain Driven Design(DDD) with the Bounded Context
- Monolithic Applications
- Challenges in Monolith Applications
- Microservices Architecture-Honey Comb Anology
- Advantages of Microservices Architecture
- Microservices Characteristics
- Microservices Principles
- Microservices Patterns
- Migration of Monolith to Microservices Architecture
- Problems in Migrating To Microservices
- Shared db?
- Communication Style synchronous/Asynchronous?
- Single JVM / Shared JVM?
- Features of Microservices
- Microservices Implementation Using Spring Boot2.x
- Securing Microservices
- Oauth2.0
- Roles in Oauth2.0
- Oauth2.0 Flow
- Terms and Grant Types
- Configuring Authorization Server
- Configuring Resource Server and Routing Tokens
- Microservices Framework and Data Management Patterns
- Axon Framework
- Components in Axon Framework
- Data Management Patterns
- Event Sourcing
- CQRS(Command Query Responsibility Segregation)
- Advantages of Event Sourcing and CQRS

Spring Cloud Content

- Spring Cloud overview
- Externalize Configurations
- problems with configurations with the microservices
- SpringCloud solution to configuration problems
- cloud config server
- cloud config client
- maintaning profiles
- install and configure RabbitMQ
- problem with refresh Endpoint
- Spring Cloud Bus
- Service Discovery at Runtime
- Eureka server Implementation
- Register a service with Eureka Server
- Service Discovery using Eureka
- Resiliency In Microservices
- Spring Cloud Hystrix
- Hystrix Circuit Breaker
- Hystrix DashBoard
- Microservices API GateWay
- Zuul Proxy as the API GateWay
- setting up zuul
- Filter in zuul
- Microservices Communication
- Feign Client
- ClientSide Load Balancer-Ribbon

e-Learning through LMS

Learning **Management System**

Our LMS (LearnPitch) is for the administration, documentation, tracking, reporting, automation, and delivery of educational courses, training programs, or learning and development programs.

Our LMS has been designed to identify training and learning gaps, using analytical data and reporting to keep you up with the class activities.

Key Features **Learning Management System**

- Live Sessions with Class recordings
- Get study material with Assignments.
- Track your curriculum covered.
- Track your class wise attendance
- Share your feedback for Trainer & Training
- Get your Training Certificate from LMS

Training Certification

Earn Your **Certificate**

Your certificate and skills are vital to the extent of jump-starting your career and giving you a chance to compete in a global space.

Croma Campus is Nasscom Certified

Croma Campus is now

NASSCOM®

Certified Member

www.cromacampus.com

Croma Campus! Reviews

"The most rewarding part of my experience has been achieving a prestigious certification in the subject that I love. Moreover, the training offered out by the specialists are of world-class and prepares out the students for corporate world. For me Croma Campus means a lot."

"By The Students For The Students"

Your Success is **Our Story**

Bharat

"I am fully satisfied with the excellent training services received by the expert staff at Croma Campus. I want to thank Croma Campus for providing me with the most innovative and affordable training services for learning all the software testing procedures and guidelines."

Ankit

"It was a lifetime experience for me to get trained by IT Experts of Croma Campus. What I liked most about the training was the consistent high-quality education, which was friendly and co-active. The placement department was also proactive, they keep me updated regarding new job opportunities and provide the grooming session to crack the interview. At last, I would like to thank all faculty members of Croma Campus for their immense help and support."

Umesh

"Without any second thought, I will give Croma Campus 10/10. Their placement department is highly proactive. I remember they started scheduling interviews for me from the very next day when I told them my course has been completed. These people are doing a phenomenal job and I highly recommend Croma Campus to everyone."

Shams Khan

"Croma Campus is doing a phenomenal job in the IT training industry. The reason why I decided to join their training program was that they provide quality training at very a nominal price. Plus, the online training mode was also a factor due to which I decided to join the training program of Croma Campus as I didn't want to attend physical classes."

Meet Our Team

Sales Team

Our Sales team is highly passionate, emphatic, positive attitude, great listening skills, ability to deliver quick solutions, and they are multitasker too. Our team always remains up-to-date about all the latest technologies and market trends. With effective communication skills, they always work to deliver the right information to customers when it is needed.

Product Team

Our product team is highly functional and collaborative working together to achieve the common outcome of designing exceptional digital experiences. Each of our members is a contributor to help us achieve success in long-run. Sitting at the high-end of technology and innovation, team helps to deliver high-end customer experiences and always comes out with a big idea as a game-changing plan.

Marketing Team

Our Marketing team works as gladiators and helps us to achieve business success in all possible ways. They are included in almost everything either it is building a brand, creating brand awareness, promoting products or services, delivering trailblazing customer experiences or increasing engagement at public forums. They are the true backbone of the Company.

Content Team

Our content team is responsible for ideation, creation, optimization, and distribution of content throughout the company. The team always starts its work with a strategy, how to create high-quality contents, and how to promote or share the content. Our in-house content team help us to produce all types of contents either they are educational content pieces, marketing content, SEO content, or any other forms too.

Customer Access Team

This is the team that has actually been taken up us from reactive state to a pro-active state. The team utilizes high-valued solutions to satisfy customers in all possible ways. It is truly said that no company can succeed if your customers are not satisfied. And our customer success team is dedicatedly working to keep all the customers satisfied and we always consider our customer feedback on priority.

HR Team

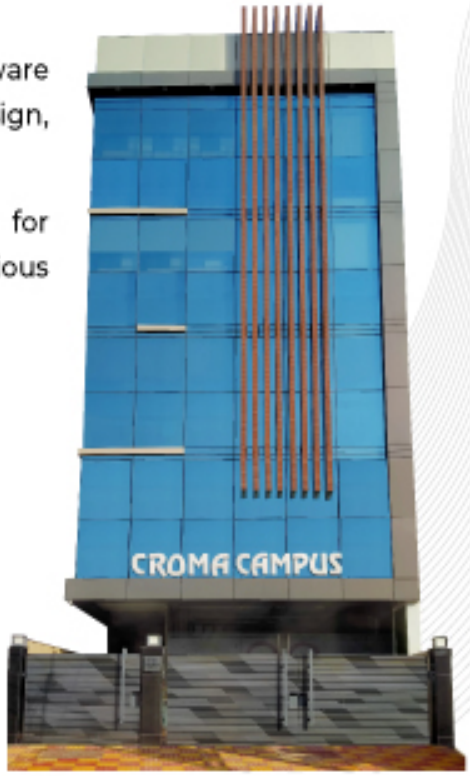
Our HR team is committed to provide high-end solutions to employees as they require. Our HR team has the right skills and knowledge to make sure that the HR department can always be legally and strategically successful. They know how to keep employees motivated all the time with the best HR policies and fun activities too from time to time.

Glimpse Of Our Office

Look **Who We are**

Our office's infrastructure comprises all the necessary software and network resources that are required to deliver IT & Design, Human Resources, Digital Marketing, and training services.

We are well-equipped with bright designed work bays for employees and managers having separate cabins with spacious cafeteria and training classrooms.



About Croma Campus

“Our Mission is to Build Nation through Education & Beyond Limitation.”

Croma Campus Training & Development Private Limited is an education platform providing rigorous industry-relevant programs designed and delivered in collaboration with world-class faculty, industry & Infrastructure. In the past 13 years we have trained 18000+ candidates and out of which we are able to place 12000+ professionals in various industries successfully.

follow us on:

cromacampus/facebook

cromacampus/pinterest

cromacampus/instagram

cromacampus/linkedin

cromacampus/twitter

cromacampus/youtube

REACH US:

Croma Campus Training & Development (P) Ltd.

G-21, Block G, Sector 3, Noida, Uttar Pradesh - 201301

+91-9711-5269-42 | +91-0120-4155255

helpdesk@cromacampus.com | www.cromacampus.com