Share on your Social Media

Java Full Stack Tutorial

Published On: July 12, 2024

Java Full Stack Tutorial

The demand for Java Full-Stack developers is increasing in the job market. Learn from the fundamentals to advanced concepts in this Java Full-Stack tutorial and get started on your career journey as a professional web developer.

Java Full Stack tutorial PDF

Introduction to Java Full-Stack

By learning Java Full-Stack, you can create dynamic web applications. It requires proficiency in Angular for developing responsive web apps, knowledge of Spring and Spring Boot for developing robust and secure back-end systems, and an understanding of RESTful web services, along with strong knowledge of HTML, CSS, and JavaScript for web development.

The following are some of the skills needed to become a proficient Java Full-Stack developer:

- **Java Programming:** Solid understanding of the fundamental ideas behind multithreading, exception management, and object-oriented programming.
- **Front-end Technologies:** Working knowledge

Featured Articles

Want to know more about becoming an expert in IT?

Click Here to Get Started

100% Placement Assurance

AUTHORISED CERTIFICATION PARTNER IBM

Quick Enquiry

Related Courses at SLA

- Java Fullstack Online Training
- Java Fullstack Training in Chennai
- Java Full Stack Developer Course

Related Posts

of HTML, CSS, JavaScript, and contemporary JavaScript frameworks such as React, Vue.js, and Angular.

- **Back-end Technologies:** The ability to create RESTful APIs using server-side technologies such as Hibernate, Spring, or Struts.
- **Relational Databases:** Proficiency with SQL and related technologies, such as MySQL, Oracle, or SQL Server, for database design, optimization, and performance tuning.
- **Web Services:** Knowledge of web service architecture, tools such as Apache Axis or Jersey, and protocols such as SOAP and REST.
- **DevOps:** Knowledge of creating, testing, and delivering applications using tools such as Docker, Jenkins, and Git.
- **Agile Methodology:** Project management experience using agile development approaches like Scrum or Kanban.

Let's explore them one by one comprehensively below.

Download Java Full Stack Syllabus PDF

Java Programming

To become a full-stack Java developer, you must be skilled in the following areas:

Core Concepts of Java

Java is a popular and flexible programming language that is well-known for its fundamental ideas that underpin software development.

To comprehend and become an expert in Java, one must grasp these fundamental ideas.

Control Flow Statements:

Three varieties of control flow statements are available in Java:

MongoDB Tutorial for Beginners

Published On: September 25, 2024

MongoDB Tutorial for Beginners In real-world contexts like e-commerce, finance, healthcare, journalism, and social networks,...

Mobile App Development Tutorial

Published On: September 25, 2024

Mobile App Development Tutorial Businesses looking to grow and scale can benefit greatly from mobile...

Microsoft Azure Tutorial for Beginners

Published On: September 24, 2024

Microsoft Azure Tutorial for Beginners We can use a wide range of services from Microsoft...

Decision-Making Statement: Java decision-making can be implemented to run a specific code segment based on the boolean result of certain circumstances stated. It includes the following:

- If
- If-else
- Nested-if
- If-else-if
- Switch cases.

Loop Commands: To repeatedly iterate over a specific code snippet, use the following:

- For Loop
- While Loop
- Do-While Loop

Jump Over Statement: Some of the statements are utilized to move past code snippets. They are,

- Break
- Continue
- Return

Methods

In Java, a method is a collection of statements that can be called whenever necessary to carry out a certain activity. The use of methods ensures code efficiency and reusability.

A full-stack Java developer should be meticulous when it comes to method declarations.

It includes the following

- Return Types
- Parameter Lists
- Exception Lists
- Method Types (predefined and user-defined)

Object-Oriented Concepts

Because Java is an object-oriented language, it employs objects as the fundamental building block to implement code or application functionality.

MEAN Stack Tutorial for Beginners
Published On: September 24, 2024

MEAN Stack Tutorial for Beginners 'M' refers to MongoDB, 'E' for Express, 'A' for Angular,...

One of the most crucial steps in becoming a full-stack Java developer is to gain a thorough understanding of object-oriented concepts, which are as follows:

- Classes
- Objects
- Abstraction
- Encapsulation
- Inheritance
- Polymorphism
- Interface
- Method Passing
- Constructors and Destructors.

Array

A Java array is a collection of variables of the same kind. These arrays are kept in infectious memory blocks and are dynamically allocated.

An array's initialization, searching, sorting, and deletion are the fundamental actions that any full-stack Java developer should know how to perform.

Framework Collection

Java's collection framework is a framework for managing and storing collections of objects.

To create basic Java programs, one must be familiar with the classes in these collections as well as their interfaces and methods.

The classes are as basic as Stack, Queue, LinkedList, and ArrayList, or as sophisticated as Set, Dequeue, Map, and so forth.

Generics

Type safety is one of the key notions to be covered in Java's foundational concepts, and generics make this possible.

It should also cover wildcards, comparable interfaces, bound type parameters, and generic methods.

Multithreading Concepts

Java's multithreading functionality is crucial since it allows two or more threads or program segments to run concurrently, ensuring optimal CPU use.

A full-stack Java developer should be proficient in the following areas:

- Using the volatile keyword
- Wait and Notify
- Interrupting
- Joining
- Thread Pools
- CountDownLatch
- Blocking Queue
- Reentrant Lock
- Deadlock.

Variations in thread creation include extending the thread class and implementing a runnable interface.

Lambda Expressions

The programs are executed using lambda expressions, which makes them shorter.

Lambda expressions allow us to write function declarations directly into classes, replacing the need for earlier classes to implement functions found in interfaces.

For a thorough understanding of lambda expressions, you must learn how to create threads using lambda expressions with variables, iterations, and predicates.

File Handling

For handling files, Java offers a robust File class. Since file management allows us to read input from files and store our output in a file, it is a crucial component.

Data streams, writing to files, making directories, reading files with bufferReader and scanners,

deleting files, and serializing objects are examples of fundamental principles.

Debugging

The practice of locating possible or current flaws in your Java code is known as debugging. Improving the quality and reusability of code is an essential step.

As a Java full-stack developer, you must understand how variables change in runtime and how to use exceptional and conditional breakpoints to efficiently debug code.

Exception Handling

Java offers two methods for handling exceptions. The try-catch-finally method can be employed to manage various exceptions. Alternatively, you may employ the "try-with-resource" strategy, which makes resource cleanup simpler.

Try, catch, finally, throw, and throws are the five keywords that form the core of Java's exception management system. The handling of exceptions is based on these terms.

Java Full Stack Inteview Questions

Front-End Technologies

A thorough understanding of front-end technologies is essential for full-stack engineers. They need to be proficient programmers in multiple languages and possess a solid grasp of how various web technologies interact.

HTML

The common markup language used to create web pages is HTML. To design user interfaces that are efficient and simple to use, a full-stack developer has to have a solid grasp of HTML.

CSS

HTML markup can be styled using CSS, a style sheet language. A full-stack developer needs to be well-versed in CSS to make visually appealing and intuitive applications and websites.

JavaScript

Programming languages like JavaScript are used to make interactive websites. A full-stack developer can only build dynamic, user-friendly websites and applications by having a solid grasp of JavaScript.

Angular.JS

One of the primary uses for AngularJs, an open-source JavaScript front-end technology, is single-page web applications (SPAs).

- It is an ever-expanding framework that offers improved methods for creating online apps. The static HTML is converted to dynamic HTML.
- Being an open-source project, anyone is free to use and modify it. Data is connected with HTML, and directives are used to extend HTML attributes.

Node.js

You can run JavaScript code outside of a browser with the help of the Node.js runtime environment. A developer can use one of several programming languages to design the backend. However, JavaScript and Node.js are widely used.

MongoDB

Data for websites and applications is frequently stored in databases like MongoDB.

For full-stack developers to build efficient and productive websites and applications, they need to have a solid grasp of databases.

Vue.js

Vue.js is an open-source JavaScript framework for creating front-end models, views, and single-page apps.

React.js

React is a JavaScript library for creating declarative, effective, and adaptable user interfaces. ReactJS is an open-source front-end library that works with components and handles only the view layer of an application. Facebook looks after it.

Bootstrap

To create responsive websites and web apps, utilize the Bootstrap toolkit, which is available for free and open source. It's the most popular HTML, CSS, and JavaScript framework for building responsive, mobile-first websites.

jQuery

An open-source JavaScript library called jQuery makes it easier for JavaScript and HTML/CSS documents.

More specifically, the Document Object Model (DOM) to interact with one another.

To clarify, jQuery makes it easier to create cross-browser JavaScript applications, handle browser events, create DOM animations, and navigate and manipulate HTML documents.

Java Full Stack Developer Salary

Back-end Technologies

The creation of an application's client-side (or server-side) is known as backend development. Understanding APIs, databases, servers, and apps is necessary for this procedure.

APIs, which are essential to backend development, manage the process of getting data from the backend and sending it to the front end.

RESTful APIs

Modern web and mobile applications are built on RESTful services and APIs, which allow disparate

systems to interact and exchange data via the Internet.

- Effective RESTful service creation for Java backend developers requires a grasp of HTTP protocols, resource naming, statelessness, and content negotiation.
- Tools and frameworks like Spring MVC help build RESTful APIs that are scalable and maintainable.
- It's also crucial to understand how to protect APIs using JWT and OAuth, two popular authentication and authorization protocols.
- For developers to facilitate smooth integration and communication between various components of an application and outside services, mastery of RESTful API development is a prerequisite.

Spring and Spring Boot

The Spring Framework, which offers a thorough programming and configuration architecture, is crucial for the creation of enterprise Java applications.

- With its embedded servers, auto-configuration, and large beginning kits, it makes it simpler to swiftly develop stand-alone, production-grade apps.
- To create reliable, effective backend systems, Java developers should become proficient with Spring and Spring Boot.
- Simple web apps and intricate, high-performance microservices ecosystems are only two examples of the many application needs that these frameworks can accommodate.

With its convention-over-configuration methodology, Spring Boot, an extension of Spring, streamlines application setup and frees up developers to concentrate on functionality rather than boilerplate code.

Gaining proficiency with Spring can greatly increase a developer's output and present a wealth of options within the Java community.

Hibernate

A framework for mapping object-oriented domain models to relational databases for web applications is provided by the open-source object-relational mapping (ORM) program Hibernate.

The abstraction that allows for object containerization and mapping is the foundation of object-relational mapping.

Struts

An open-source framework called Struts uses an MVC (Model, View, Controller) architecture and extends the Java Servlet API.

It lets you build online applications that are flexible, expandable, and maintainable using common technologies like JavaBeans, JSP pages, resource bundles, and XML.

Java Full Stack Training

Relational Databases

The foundation of full-stack development is relational database management, which makes it easier to store, organize, and retrieve the data that drives applications.

It plays a crucial role in guaranteeing data security, integrity, and performance, which directly affects the finished product's functionality and user experience.

SQL

The acronym for Structured Query Language is SQL. The majority of database queries for adding, removing, or modifying data are based on the normal SQL syntax, except for a few small syntax

modifications.

MySQL

One of the first open-source RDBMSs to be created and released was MySQL. There are numerous MySQL variations at the moment.

- MySQL can be used with all major operating systems and was created and implemented in the C and C++ programming languages.
- It is a fundamental part of the widely used LAMP (Linux, Apache, MySQL, PHP/Perl/Python) open-source web application software stack.

SQL Server

Microsoft created SQL Server, a potent relational database management system.

- It offers a platform for data management, retrieval, and storage for a range of services and applications.
- SQL is a computer language that is used to interact with databases, and SQL Server supports it.
- It enables users to execute many tasks, including querying, sorting, filtering, and joining data, as well as build, edit, and remove databases, tables, and views.
- High availability, security, and scalability are just a few of the sophisticated features that make SQL Server a popular option for companies of all kinds.
- Web applications, corporate intelligence, and data analysis frequently use it.

Web Services Architecture

The Web Services architecture outlines how to interoperablely implement the actions and instantiate the pieces.

The three roles of service provider, service requester, and service registry interact within the architecture of a web service.

The three operations, such as publish, find, and bind, are involved in the interaction.

These roles and operations affect the web service artifacts. The web service software module and its description make up the web service artifacts.

SOAP

Simple Object Access Protocol is abbreviated as SOAP. It outlines the XML standard format. It also outlines the methodology for developing web services. The format of the request XML and the return XML are defined using Web Service Definition Language (WSDL).

REST

'REpresentational State Transfer' is referred to as REST. Roy Thomas Fielding, who also created HTTP, is the one behind its development.

Increasing the effectiveness of web services is the primary objective of RESTful web services.

Web services that are RESTful attempt to design services utilizing the many notions that are already part of HTTP. REST is not a protocol; it is an architectural method.

Apache Axis

Developers may design distributed computing apps that are compatible by using Apache Axis.

It includes several tools and APIs for creating and distributing Web service applications, as well as implementations of the SOAP server in Java and C++.

Jersey

An open-source framework called Jersey helps Java programmers create RESTful web services. It implements the Java API for RESTful Web Services (JAX-RS) reference model, which offers a standardized approach to web service development with Java.

DevOps

An increasingly important component of the skill set of full-stack developers is the adoption of DevOps processes and tools. Understanding cloud computing platforms, pipelines for continuous integration and deployment (CI/CD), and containerization technologies like Docker and Kubernetes are all part of this.

Among the fundamental ideas of DevOps are:

Culture: Dismantling conventional silos and fostering a culture of cooperation and shared accountability between the development and operations teams.

Automation: Automating repetitive jobs to improve productivity, lower mistake rates, and guarantee consistent development, testing, and deployment procedures.

Lean Thinking: Applying lean concepts to enhance productivity, reduce waste, and pursue ongoing development.

Measurement: To inform choices and make improvements, application performance, and user experience are continuously measured.

Sharing: Teams should share techniques and expertise to improve learning and creativity,

Docker

Software is delivered in packages known as containers using OS-level virtualization via a group of platform-as-a-service products named Docker.

Jenkins

Open source automation server Jenkins is available. It facilitates continuous integration and continuous delivery by helping automate the portions of **software development** that are linked to building, testing, and deploying.

Git

Git is a file version tracking distributed version control system. While working together to develop software, programmers frequently utilize it to manage source code.

Agile

Agile software development is an incremental and iterative software development methodology that prioritizes frequent and rapid delivery of a usable result.

To make sure the product satisfies the customer's wants and expectations, the development team and the customer work closely together.

It contains the following process:

- **Requirements gathering:** Prioritizing is done after gathering the customer's software requirements.
- **Planning:** The development team plans out the features that will be included in each program iteration.
- **Development:** With frequent and quick iterations, the development team builds the software.
- **Testing:** The program is put through a rigorous testing process to make sure it is high-quality and fits the needs of the user.
- **Deployment:** The program is implemented and made available.
- **Maintenance:** To guarantee that it continues to satisfy the needs and expectations of the user, the program is maintained.

Conclusion

We hope this Java full-stack tutorial is useful to you in gaining expertise with the required skills. Learn comprehensively with hands-on exposure in our **Java Full-Stack training in Chennai.**

Share on your Social Media

Softlogic Academy

Softlogic Systems

KK Nagar [Corporate Office]

No.10, PT Rajan Salai, K.K. Nagar, Chennai – 600 078.
Landmark: Karnataka Bank Building
Phone: +91 86818 84318
Email: enquiry@softlogicsys.in
Map: Google Maps Link

OMR

No. E1-A10, RTS Food Street
92, Rajiv Gandhi Salai (OMR),
Navalur, Chennai - 600 130.
Landmark: Adj. to AGS Cinemas
Phone: +91 89256 88858
Email: info@softlogicsys.in
Map: Google Maps Link

Navigation

- About Us
- Blog Posts
- Careers
- Contact
- Placement Training
- Corporate Training
- Hire With Us
- Job Seekers
- SLA's Recently Placed Students
- Reviews
- Sitemap

Important Links

- Disclaimer
- Privacy Policy
- Terms and Conditions

Courses

- Python
- Software Testing
- Full Stack Developer
- Java
- Power BI
- Clinical SAS

Social Media Links

Review Sources

- Google
- Trustpilot

- Data Science
- Embedded
- Cloud Computing
- Hardware and Networking
- VBA Macros
- Mobile App Development
- DevOps

- Glassdoor
- Mouthshut
- Sulekha
- Justdial
- Ambitionbox
- Indeed
- Software Suggest
- Sitejabber

Copyright © 2024 - Softlogic Systems. All Rights Reserved

SLA™ is a trademark of Softlogic Systems, Chennai. Unauthorised use prohibited.