



Java Full Stack Development

Course Brochure

To explore other learning opportunities, please visit

<http://www.bskilling.com/>

or write to us at

support@bskilling.com

Java Full Stack Development

Price: 81999 (INR)

Level: Advanced

Training Type: Online

Duration: 180 Hrs

In Java full-stack development, the body of the project is the application's core, encompassing code, modules, and components for front-end and back-end systems. It involves server-side logic, web page creation, database integration, business rules, user input, and data management. The body is where development work shapes the application's functionality and behavior to meet requirements.

Overview

Java full-stack development utilizes Java for both front-end and back-end of web applications. Java developers handle the entire development process, from designing the user interface to data storage. Front-end uses HTML, CSS, JavaScript for interactive web pages. Back-end involves Java code for server-side processing, testing for quality assurance, and deployment for user accessibility. Java full-stack enables robust and scalable web applications, meeting diverse user needs effectively.

Objectives

- Develop web applications using Java for both front-end and back-end.
- Create a web application that uses Java for both the user interface (UI) and the back-end logic.
- Use Java frameworks and libraries to streamline development and enhance efficiency.
- Develop web applications using Java for both front-end and back-end.
- Create a web application that uses Java for both the user interface (UI) and the back-end logic.
- Use Java frameworks and libraries to streamline development and enhance efficiency.

Prerequisites

- Basic programming concepts.
- Proficiency in Java programming language. Familiarity with HTML, CSS, and JavaScript.
- Understanding of relational databases and SQL.
- Set up Java development environment with JDK and IDE.

- Knowledge of server-side technologies like servlets, JSP, or JSF.
- Familiarity with Java frameworks like Spring Boot or Java EE.

Skills Covered

- Proficiency in Java programming
- Front-end development skills (HTML, CSS, JavaScript)
- Back-end development skills (Java servlets, server-side programming)
- Database management skills (SQL, JDBC, ORM frameworks)
- Knowledge of Java frameworks (Spring Boot, Java EE)
- Understanding of web services (RESTful APIs, SOAP)

Audience

- Aspiring developers
- Experienced Java developers
- Web developers
- Software engineers
- Computer science students
- Professionals seeking a career transition

Key Features

- Full-stack development capability
- Java programming language
- Robust frameworks (e.g., Spring Boot, Java EE)
- Scalability for handling increasing user loads
- Seamless database integration (JDBC, ORM frameworks)
- Web services support (RESTful APIs, SOAP)

Resources

- Online training courses
- reference materials
- Practice exams Study guides and

Benefits

- Strong community support and extensive resources

- Efficient and streamlined development process
- Reusability of code components
- Enhanced productivity with frameworks and libraries
- Versatility and flexibility of the Java language
- Scalability and ability to handle growing user demands

Curriculum

Introduction to Java Programming:

1. Basics of Java syntax, variables, data types, and control structures.
2. Object-oriented programming (OOP) concepts, including classes, objects, inheritance, and polymorphism.

Front-End Development:

1. HTML: Markup language for creating web pages.
2. CSS: Styling and layout of web pages.
3. JavaScript: Client-side scripting for interactivity and dynamic behavior.

Back-End Development:

1. Java Servlets: Handling HTTP requests and generating dynamic responses.
2. JavaServer Pages (JSP) or JavaServer Faces (JSF): Server-side rendering of web pages.
3. Spring Boot or Java EE: Java frameworks for building scalable and modular back-end applications.
4. Database integration: JDBC for interacting with databases, Hibernate for ORM.

Web Services and APIs:

1. SOAP-based web services: Working with SOAP protocols and frameworks.
2. RESTful APIs: Building and consuming web services using Java.

Testing and Quality Assurance:

1. Unit testing with JUnit or TestNG.
2. Integration testing for testing interactions between components.
3. Test automation frameworks like Selenium for web application testing.

Version Control and Collaboration:

1. Git: Version control system for managing codebase and collaboration.
2. GitHub or GitLab: Hosting platforms for code repositories and collaborative

development.

Deployment and DevOps:

1. Application servers like Apache Tomcat or WildFly for hosting Java applications.
2. Build tools like Apache Maven or Gradle for dependency management and build automation.
3. Continuous Integration/Continuous Deployment (CI/CD) using tools like Jenkins or GitLab CI.

Security:

1. Web application security principles and best practices. Handling user authentication and authorization.
2. Protection against common security vulnerabilities.

Advanced Topics:

1. Performance optimization techniques.
2. Caching mechanisms for improving application performance.
3. Microservices architecture and containerization with technologies like Docker.

Outcomes

- Development of robust and scalable web applications using Java
- Seamless integration of front-end and back-end components
- Delivery of high-quality applications through thorough testing
- Successful deployment and availability of the web application
- Improved career prospects and job opportunities in Java full-stack development
- Creation of user-friendly and visually appealing user interfaces

Certification

Join us now to accelerate your career with industry-recognized credentials and stay ahead in the ever-evolving world of cloud technology.

FAQs

1. What are some popular Java frameworks for full-stack development?

Some popular Java frameworks for full-stack development include Spring Boot, Java EE, Apache Struts, and Play Framework. These frameworks provide a comprehensive set of tools,

libraries, and abstractions to simplify development tasks, enhance productivity, and facilitate the integration of front-end and back-end components.

2. Can I use Java for both front-end and back-end development?

Yes, Java can be used for both front-end and back-end development. While Java is primarily known for its back-end capabilities, with the advent of frameworks like JavaServer Faces (JSF) and JavaServer Pages (JSP), it is possible to use Java for server-side rendering of web pages and implementing front-end functionality alongside HTML, CSS, and JavaScript.

3. What skills are important for a Java full-stack developer?

Important skills for a Java full-stack developer include proficiency in Java programming, knowledge of front-end technologies like HTML, CSS, and JavaScript, experience with Java frameworks like Spring Boot or Java EE, understanding of databases and SQL, familiarity with web services and APIs, testing skills, and knowledge of version control systems like Git.

4. Is Java full-stack development in demand?

Yes, Java full-stack development is in high demand in the industry. Java is widely used for enterprise-level web applications, and companies are constantly seeking skilled developers who can work on both the front-end and back-end components. Job opportunities for Java full-stack developers are plentiful, making it a promising career path in the software development field.

5. What is the role of Java in full-stack development?

Java is a versatile programming language used in full-stack development to build both the front-end and back-end components of web applications. It allows developers to create user interfaces, handle server-side logic, interact with databases, and integrate various components of the application.

6. What are the advantages of using Java for full-stack development?

Java offers several advantages in full-stack development, including its robustness, scalability, extensive framework support, cross-platform compatibility, strong community support, and availability of libraries and tools. Java's object-oriented nature and rich ecosystem make it well-suited for building complex and feature-rich web applications.