

Java Full Stack

Course outline

Module 1: Introduction to Java

Module 1: Introduction to Java is an introductory course designed to provide students with a comprehensive overview of the Java programming language. This module covers the fundamentals of the language, including variables, data types, operators, control flow, classes, objects, and more. It also introduces the Java Development Kit (JDK) and the Eclipse IDE, which are essential tools for developing Java applications.

Lessons

- Overview of Java and its features
- Setting up the development environment
- Understanding the Java syntax
- Working with variables and data types
- Working with operators and expressions
- · Working with control flow statements
- · Working with classes and objects
- Working with inheritance and polymorphism
- Working with packages and interfaces
- · Working with exceptions and errors
- Working with collections and generics
- Working with threads and concurrency
- Working with I/O streams
- Working with JDBC and databases
- Working with JavaFX and GUI programming

After completing this module, students will be able to:

- Understand the fundamentals of the Java programming language
- Write basic Java programs using variables, operators, and control flow
- Utilize the Java API to create classes, objects, and methods
- Debug and troubleshoot Java code to identify and fix errors

Module 2: Object-Oriented Programming with Java

Module 2 of the Java Full Stack course focuses on Object-Oriented Programming with Java. It covers topics such as classes, objects, inheritance, polymorphism, and interfaces. Students will learn how to create and use classes, create and use objects, and use inheritance and polymorphism to create more efficient and reusable code. Additionally, students will learn how to use interfaces to create more flexible

and extensible code.

Lessons

- Introduction to Object-Oriented Programming
- Classes and Objects
- Inheritance and Polymorphism
- Interfaces and Abstract Classes
- Exception Handling
- Generics
- Collections Framework
- Streams and Lambda Expressions
- Multithreading
- GUI Programming with JavaFX

After completing this module, students will be able to:

- Understand the fundamentals of object-oriented programming and the Java language.
- Develop and implement classes, objects, and methods in Java.
- Utilize the Java API to create and manipulate objects.
- Create and debug programs using the Eclipse IDE.

Module 3: Java Collections Framework

Module 3 of the Java Full Stack course covers the Java Collections Framework, which provides a set of classes and interfaces for managing collections of objects. It includes topics such as the List, Set, and Map interfaces, as well as the ArrayList, LinkedList, HashSet, and TreeSet classes. Students will learn how to use these classes and interfaces to store, retrieve, and manipulate data in an efficient and effective manner.

- Overview of the Java Collections Framework
- Working with the Collection Interfaces
- Implementing the Collection Interfaces
- Working with the List Interface
- Working with the Set Interface
- Working with the Map Interface
- Working with the Queue Interface
- Working with the Deque Interface
- · Working with the SortedSet Interface
- Working with the SortedMap Interface
- Working with the PriorityQueue Interface
- Working with the NavigableSet Interface
- Working with the NavigableMap Interface
- Working with the Iterator Interface
- Working with the ListIterator Interface
- Working with the Spliterator Interface

- Working with the Comparator Interface
- Working with the Comparable Interface
- Working with the Arrays Class
- Working with the Collections Class

- Understand the fundamentals of the Java Collections Framework and its components.
- Utilize the Java Collections Framework to create and manipulate collections of objects.
- Implement efficient algorithms to search, sort, and traverse collections.
- Develop applications that use the Java Collections Framework to store and manipulate data.

Module 4: Java Database Connectivity (JDBC)

Module 4 of the Java Full Stack course covers Java Database Connectivity (JDBC). This module provides an introduction to the fundamentals of JDBC, including how to connect to a database, execute queries, and process results. It also covers advanced topics such as transactions, stored procedures, and batch processing. By the end of this module, students will have a solid understanding of how to use JDBC to interact with databases in Java.

Lessons

- Introduction to JDBC
- Establishing a Connection to a Database
- Executing SQL Statements
- Working with Result Sets
- Prepared Statements
- Transactions
- Stored Procedures
- Database Metadata
- Error Handling
- Optimizing Performance with JDBC

After completing this module, students will be able to:

- Understand the fundamentals of JDBC and its components
- Develop applications that interact with a database using JDBC
- Create and execute SQL statements using JDBC
- Utilize JDBC to connect to a database and perform CRUD operations

Module 5: Java Servlets and JavaServer Pages (JSP)

Module 5 of the Java Full Stack course covers the fundamentals of Java Servlets and JavaServer Pages (JSP). Students will learn how to create dynamic web applications using the Java Servlet and JSP technologies. Topics include the basics of servlet and JSP programming, the servlet life cycle, the JSP life cycle, and the use of custom tags. Additionally, students will learn how to use the JavaServer Pages Standard Tag Library (JSTL) to create powerful web applications.

Lessons

- Introduction to Java Servlets
- Configuring and Deploying Java Servlets
- Working with JavaServer Pages (JSP)
- Developing JSP Pages with JavaBeans
- Using JSP Standard Tag Library (JSTL)
- Implementing MVC Architecture with Java Servlets and JSP
- Working with Java Filters
- Securing Java Web Applications
- Debugging and Troubleshooting Java Servlets and JSP
- Performance Tuning of Java Servlets and JSP

After completing this module, students will be able to:

- Understand the fundamentals of Java Servlets and JavaServer Pages (JSP)
- Develop web applications using Java Servlets and JavaServer Pages (JSP)
- Implement the Model-View-Controller (MVC) design pattern in web applications
- Utilize the Java EE platform to create dynamic web applications with Java Servlets and JavaServer Pages (JSP)

Module 6: Java Web Services

Module 6 of the Java Full Stack course focuses on Java Web Services. Students will learn how to create and deploy web services using Java technologies such as JAX-WS, JAX-RS, and SOAP. They will also learn how to consume web services from other applications and how to secure web services. Finally, they will learn how to use web services to build distributed applications.

Lessons

- Introduction to Web Services
- Understanding SOAP and RESTful Web Services
- Developing SOAP Web Services
- Developing RESTful Web Services
- Securing Web Services
- Consuming Web Services
- Testing Web Services
- Debugging Web Services
- Deploying Web Services
- Troubleshooting Web Services

- Understand the fundamentals of web services and how to create and consume them using Java.
- Develop and deploy web services using Java technologies such as JAX-WS, JAX-RS, and SOAP.
- Create and deploy web services using popular frameworks such as Apache CXF and Spring.

• Utilize web services to integrate applications and services across different platforms.

Module 7: Java Enterprise Edition (JEE)

Module 7 of the Java Full Stack course covers the Java Enterprise Edition (JEE) platform. It provides an overview of the JEE architecture, its components, and how to use them to develop enterprise applications. It also covers topics such as web services, security, and distributed computing. This module is essential for anyone looking to develop enterprise applications using Java.

Lessons

- Introduction to JEE Architecture
- Developing Web Applications with JEE
- Working with Java Servlets
- Working with JavaServer Pages (JSP)
- Working with JavaServer Faces (JSF)
- Working with Enterprise JavaBeans (EJB)
- Working with Java Persistence API (JPA)
- Working with Java Message Service (JMS)
- Working with Java Transaction API (JTA)
- Working with JavaMail API
- Working with Java Naming and Directory Interface (JNDI)
- Working with Java Authentication and Authorization Service (JAAS)
- Working with Java Authorization Contract for Containers (JACC)
- Working with Java Management Extensions (JMX)
- Working with Java Web Services
- Working with Java Security
- Working with Java Connector Architecture (JCA)
- Working with Java Management Extensions (JMX)
- Working with JavaServer Pages Standard Tag Library (JSTL)
- Working with JavaServer Faces (JSF) Tag Libraries

After completing this module, students will be able to:

- Understand the fundamentals of Java Enterprise Edition (JEE) and its components.
- Develop and deploy enterprise applications using JEE technologies such as JavaServer Pages (JSP), JavaServer Faces (JSF), Enterprise JavaBeans (EJB), and Java Persistence API (JPA).
- Utilize the Java Message Service (JMS) to create distributed applications.
- Implement web services using Java API for XML Web Services (JAX-WS) and Java API for RESTful Web Services (JAX-RS).

Module 8: Java Persistence API (JPA)

Module 8 of the Java Full Stack course covers the Java Persistence API (JPA), which is a Java application programming interface used to access and manage data stored in a relational database. This module will teach students how to use JPA to create, read, update, and delete data from a database, as well as how to use JPA to create and manage database transactions.

Lessons

- Introduction to JPA
- JPA Architecture
- Configuring JPA
- Working with Entities
- Mapping Relationships
- Querying with JPQL
- Advanced Querying with Criteria API
- Caching with JPA
- Transactions and Concurrency
- Integrating with Spring Framework

After completing this module, students will be able to:

- Understand the fundamentals of the Java Persistence API (JPA) and its components.
- Develop and implement JPA entities and relationships.
- Utilize the JPA API to perform CRUD operations on a database.
- Implement advanced JPA features such as caching, lazy loading, and query optimization.

Module 9: Java Message Service (JMS)

Module 9 of the Java Full Stack course covers the Java Message Service (JMS) API, which is a Java API for sending and receiving messages between applications. It provides a common interface for applications to communicate with each other, regardless of the underlying messaging system. This module will teach students how to use the JMS API to create, send, and receive messages, as well as how to configure and deploy JMS applications.

- Introduction to JMS
- JMS Architecture
- JMS API
- JMS Message Types
- JMS Message Delivery Modes
- JMS Message Selectors
- JMS Message Browsing
- JMS Message Filtering
- JMS Message Transformation
- JMS Message Persistence
- JMS Message Expiration
- JMS Message Correlation
- JMS Message Routing
- JMS Message Security
- JMS Message Transactions
- JMS Message Groups
- JMS Message Compression

- JMS Message Prioritization
- JMS Message Acknowledgement
- JMS Message Delivery Assurance

- Understand the fundamentals of JMS and its components.
- Develop applications using JMS API.
- Implement message-oriented middleware using JMS.
- Create and deploy message-driven beans.

Module 10: Java Server Faces (JSF)

Module 10 of the Java Full Stack course covers Java Server Faces (JSF), a popular web application framework for Java. Students will learn how to create web applications using JSF components, manage user input, and create custom components. Additionally, students will learn how to use JSF with other technologies such as AJAX, HTML, and CSS.

Lessons

- Introduction to JSF
- JSF Architecture
- JSF Components
- JSF Expression Language
- JSF Navigation
- JSF Validation
- JSF Data Conversion
- JSF Event Handling
- JSF Ajax
- JSF Custom Components
- JSF Security
- JSF Internationalization
- JSF Integration with Other Frameworks
- JSF Best Practices

After completing this module, students will be able to:

- Understand the fundamentals of JSF and its components
- Develop web applications using JSF
- Implement navigation rules and page flows
- Integrate JSF with other Java technologies such as Spring, Hibernate, and JPA

Module 11: Java Web Frameworks (Spring, Hibernate, Struts)

Module 11 of the Java Full Stack course covers the use of Java web frameworks such as Spring, Hibernate, and Struts. Students will learn how to use these frameworks to create web applications and services, as well as how to integrate them with other technologies. They will also learn how to use the frameworks to create secure and efficient web applications.

Lessons

- Introduction to Java Web Frameworks
- Overview of Spring Framework
- Dependency Injection with Spring
- Working with Spring MVC
- Introduction to Hibernate
- Object-Relational Mapping with Hibernate
- Introduction to Struts
- Working with Struts Actions and Results
- Integrating Spring and Hibernate with Struts
- Security and Authentication with Spring Security
- Testing Java Web Applications
- Deploying Java Web Applications

After completing this module, students will be able to:

- Understand the fundamentals of web application development using Java web frameworks.
- Develop web applications using Spring, Hibernate, and Struts.
- Implement the Model-View-Controller (MVC) architecture in web applications.
- Utilize the features of Spring, Hibernate, and Struts to create robust web applications.

Module 12: Java Application Servers (Tomcat, JBoss, Glassfish)

Module 12 of the Java Full Stack course covers the use of Java Application Servers such as Tomcat, JBoss, and Glassfish. Students will learn how to install, configure, and deploy applications on these servers. Additionally, students will learn how to troubleshoot and debug applications running on these servers.

Lessons

- Overview of Java Application Servers
- Installing and Configuring Tomcat
- Deploying Applications on Tomcat
- · Working with JBoss
- Deploying Applications on JBoss
- Understanding Glassfish
- Deploying Applications on Glassfish
- Troubleshooting Java Application Servers
- Security Considerations for Java Application Servers
- Performance Tuning for Java Application Servers

- Understand the architecture of Java application servers and their components.
- Configure and deploy applications on Tomcat, JBoss, and Glassfish.
- Monitor and troubleshoot application server performance.
- Develop and deploy web services on application servers.

Module 13: Java Security

Module 13 of the Java Full Stack course covers the fundamentals of Java security. It covers topics such as authentication, authorization, cryptography, and secure coding practices. It also provides an overview of the Java Security API and how to use it to secure applications.

Lessons

- Introduction to Java Security
- Authentication and Authorization
- Cryptography Basics
- Java Cryptography Architecture
- Java Security APIs
- Secure Coding Practices
- Secure Network Communications
- Secure Storage of Data
- Secure Web Applications
- Secure Mobile Applications
- Secure Cloud Computing
- Secure Database Access
- Secure Web Services
- Secure Messaging
- Secure System Administration
- Secure Software Development Lifecycle
- Secure Software Testing
- Secure Software Deployment
- Secure Software Maintenance
- Secure Software Updates

After completing this module, students will be able to:

- Understand the fundamentals of Java security and how to apply them to secure applications.
- Implement authentication and authorization techniques to protect applications from malicious attacks.
- Utilize encryption and hashing algorithms to secure data.
- Develop secure web applications using Java Security APIs.

Module 14: Java Testing (JUnit, Mockito)

Module 14 of the Java Full Stack course covers the fundamentals of Java testing, including the use of JUnit and Mockito. Students will learn how to write and execute unit tests, create mock objects, and use assertions to verify the behavior of their code. Additionally, they will gain an understanding of the

importance of testing and how it can help ensure the quality of their applications.

Lessons

- Introduction to JUnit
- Writing Unit Tests with JUnit
- Using Assertions in JUnit
- Parameterized Tests in JUnit
- Introduction to Mockito
- Writing Mocks with Mockito
- Using Mockito to Mock Dependencies
- Verifying Interactions with Mockito
- Best Practices for Unit Testing
- Automating Unit Tests with Maven

After completing this module, students will be able to:

- Understand the fundamentals of unit testing and integration testing in Java.
- Create and execute unit tests using JUnit and Mockito.
- Utilize Mockito to mock objects and verify behavior.
- Implement test-driven development (TDD) and behavior-driven development (BDD) techniques.

Module 15: Java Build Tools (Maven, Ant)

Module 15 of the Java Full Stack course covers the use of Java build tools such as Maven and Ant. Students will learn how to use these tools to automate the process of building, testing, and deploying Java applications. They will also learn how to configure and customize the build process to meet their specific needs.

Lessons

- Introduction to Java Build Tools
- Understanding Maven
- Configuring Maven Projects
- Working with Maven Plugins
- Understanding Ant
- Configuring Ant Projects
- Working with Ant Tasks
- Comparing Maven and Ant
- Automating Builds with Maven and Ant
- Troubleshooting Builds with Maven and Ant

- Understand the fundamentals of Maven and Ant build tools.
- Create and configure Maven and Ant build scripts.

- Utilize Maven and Ant to automate the build process.
- Troubleshoot and debug Maven and Ant build scripts.

Module 16: Java Continuous Integration (Jenkins)

Module 16 of the Java Full Stack course covers the use of Jenkins for Continuous Integration (CI) in Java. Students will learn how to set up and configure Jenkins for CI, as well as how to use Jenkins to automate the build, test, and deployment of Java applications. Additionally, students will learn how to use Jenkins to monitor and report on the performance of their applications.

Lessons

- Introduction to Jenkins
- Setting up Jenkins for Java Projects
- Configuring Jenkins for Continuous Integration
- Automating Builds with Jenkins
- Integrating Jenkins with Source Control
- Automating Tests with Jenkins
- Deploying Applications with Jenkins
- Monitoring Jenkins Performance
- Troubleshooting Jenkins
- Best Practices for Jenkins

After completing this module, students will be able to:

- Understand the concepts of Continuous Integration (CI) and Continuous Delivery (CD)
- Set up and configure Jenkins for CI/CD
- · Create and configure Jenkins jobs for automated builds and deployments
- Utilize Jenkins plugins to extend the functionality of Jenkins for CI/CD

Module 17: Java Debugging and Profiling

Module 17 of the Java Full Stack course focuses on debugging and profiling techniques for Java applications. Students will learn how to use the Java Debugger and Profiler to identify and fix errors in their code, as well as how to optimize their code for better performance. This module will also cover topics such as memory management, threading, and garbage collection.

- Introduction to Debugging in Java
- Debugging Techniques in Java
- Debugging Tools in Java
- Debugging Exceptions in Java
- Introduction to Profiling in Java
- Profiling Tools in Java
- Profiling Performance in Java
- Analyzing Memory Usage in Java

- Analyzing Threads in Java
- Analyzing Garbage Collection in Java

- Understand the fundamentals of debugging and profiling in Java.
- Utilize the debugging and profiling tools available in Java.
- Identify and troubleshoot errors in Java code.
- Optimize the performance of Java applications through debugging and profiling.

Module 18: Java Design Patterns

Module 18 of the Java Full Stack course covers the fundamentals of Java Design Patterns. Students will learn the different types of design patterns, their benefits, and how to apply them to their own projects. They will also gain an understanding of the principles of object-oriented programming and how to use them to create efficient and maintainable code.

- Introduction to Design Patterns
- Creational Design Patterns
- Structural Design Patterns
- Behavioral Design Patterns
- Singleton Pattern
- Factory Pattern
- Abstract Factory Pattern
- Builder Pattern
- Prototype Pattern
- Adapter Pattern
- Bridge Pattern
- Composite Pattern
- Decorator Pattern
- Facade Pattern
- Flyweight Pattern
- Proxy Pattern
- Chain of Responsibility Pattern
- Command Pattern
- Interpreter Pattern
- Iterator Pattern
- Mediator Pattern
- Memento Pattern
- Observer Pattern
- State Pattern
- Strategy Pattern
- Template Method Pattern
- Visitor Pattern

- Understand the fundamentals of design patterns and their application in Java.
- Implement the most commonly used design patterns in Java.
- Analyze the advantages and disadvantages of different design patterns.
- Utilize design patterns to create efficient and maintainable code.

Module 19: Java Cloud Computing (AWS, Azure, Google Cloud Platform)

Module 19 of the Java Full Stack course covers the fundamentals of cloud computing using the three major cloud providers: Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform. Students will learn how to deploy and manage applications in the cloud, as well as how to use cloud services such as storage, databases, and analytics. Additionally, students will gain an understanding of the various pricing models and security considerations associated with cloud computing.

Lessons

- Introduction to Cloud Computing
- Setting up a Cloud Environment
- Working with AWS Services
- Working with Azure Services
- Working with Google Cloud Platform Services
- Deploying Java Applications to the Cloud
- Managing Cloud Resources
- Security and Compliance in the Cloud
- Automating Cloud Infrastructure
- Monitoring and Troubleshooting Cloud Applications

After completing this module, students will be able to:

- Understand the fundamentals of cloud computing and the different cloud service providers (AWS, Azure, Google Cloud Platform).
- Develop and deploy applications on the cloud using Java.
- Utilize cloud services such as storage, databases, and analytics.
- Monitor and manage cloud resources and services.

Module 20: Java DevOps (Docker, Kubernetes)

Module 20 of the Java Full Stack course focuses on DevOps for Java applications. It covers topics such as Docker and Kubernetes, which are essential for deploying and managing Java applications in production. Students will learn how to use these tools to create and manage containers, deploy applications, and monitor their performance.

Lessons

• Introduction to Docker and Kubernetes

- Setting up a Docker Container
- Working with Kubernetes
- Deploying Java Applications with Docker and Kubernetes
- Automating Deployment with Kubernetes
- Monitoring and Logging with Kubernetes
- Security and Access Control with Kubernetes
- Troubleshooting and Debugging with Kubernetes
- Best Practices for Java DevOps with Docker and Kubernetes

- Understand the fundamentals of Docker and Kubernetes and how they are used in DevOps.
- Create and deploy Docker containers and Kubernetes clusters.
- Utilize Docker and Kubernetes to automate the deployment of Java applications.
- Monitor and troubleshoot Java applications running in Docker and Kubernetes environments.