Unogeeks

Java Certification Training Course Content

Course Duration – 45-50 Hrs., 7-8 Weeks

Course Information

Batch Options

Weekday Batch
Mon – Fri – 1.5 Hr./Day

About the Trainer

Industry Expert Trainer with 15+ Years

Real Time Work Experience at Top US Based Product and Consulting Firms

Contact Us

Mobile:	+91 73960 33555
Whatsapp:	+91 73960 33555
Mail:	Prasad@unogeeks.com
Website:	Unogeeks.com

Introduction To Java Training

Java

Java is a popular programming language. Java is used to develop mobile apps, web apps, desktop apps, games and much more.

What you'll learn

- Core Java - Data Types, Methods, Classes, Packages, CLA and OOPS
- Advanced Java Skills - Multi-Threading, Garbage Collection, & IO Streams
- Strings, Collections and Networking
- Advanced Java - JDBC, Servlets, & JSP
- Master Spring Framework & Core Modules, JDBC, ORM, & AOP
- Spring Transaction Management, MVC and JEE
- Build 2 Complex Realtime Java Projects
- Prepare for Java certification exams and get Job Ready
- Resume & Interview preparation and Job Assistance

Course Content

********* PART 1: Core Java *********

Module 1: Introduction

- Introduction to computer languages?
- Need of computer languages
- Java v/s other languages Platform dependencies?
- Introduction to Java
- History of Java JDK software
- Platform Independency
- Structure of Java-application

Module 2: Java Basics

- Introduction to variables and identifiers
- Type of variables
- Memory construction and destruction of variables
- Rules to create variables
- Naming conventions of variables
- Introduction to Blocks and Methods

Module 3: Data Types and Methods

- Primitive types
- Reference types
- Introduction to Java Methods
- Need & Classification of functions

Module 4: Build First Java App

- Class keyword and its naming convention
- Class naming convention
- Why main has public access
- Why main has static modifier
- Why return type is void
- System.out.println() description
- Naming convention of methods and packages.
- Overloading main method
- Static members (class level)
- Non-static members (object level)

Module 5: Wrapper classes

- Introduction
- Why wrapping
- Boxing and Unboxing
- Auto Boxing and Auto Unboxing
- Primitive to String conversion and vice-versa
- Usage of access modifiers in packages

Module 6: Packages

- Introduction to Packages
- How to create user defined package
- Calling members of same and different packages
- Import statement
- Need of fully qualified name
- Difference between import and include
- Creating and Importing Sub packages

Module 7: Command Line Argument

- Introduction to CLA
- Advantage and disadvantages of CLA
- Parse methods
- Scanner and Random classes

Module 8: OOPS

- Introduction to OOPS
- Classes and Object
- Encapsulation and Abstraction
- Inheritance and final keyword
- Polymorphism
- Abstract classes
- Interfaces, Aggregation, Association & Composition
- Singleton and Factory classes

Module 9: Multi-Threading

- Introduction to multi-tasking and multi-threading
- Drawbacks in multi-tasking
- Creation of Thread
- Life cycle of Thread
- Threads Using Thread and Runnable interfaces
- Time management in multi-tasking and multi-threading
- Sleep(), join(), wait(), notify(), notifyAll()
- Naming to threads via constructors or via setters
- Constructors of Thread class

Module 10: Garbage Collection

- Introduction GC
- Introduction to daemon threads
- Drawbacks of
- finalize(), gc(), exec()
- JVM memory increment
- Process class

Module 11: IO Streams

- Byte & Character streams
- Object streams (Serialization)
- Why serialization
- Transient keyword
- What is persistent state of an object
- Marker interface
- Can we serialize subclass object if super class is not Serializable
- Can we serialize an object contains reference to non-Serializable object
- Byte array streams, Data input streams, Buffered streams
- Working with files

Module 12: Reflection API and Inner Classes

- Static inner classes
- Non-static inner classes
- Local inner classes
- Anonymous inner classes

Module 13: Strings

- Introduction to Strings
- Creating objects to String
- String library functions
- Mutable objects
- Immutable objects
- String/String Buffer/String Reader
- Creating Immutable class

Module 14: Networking

- Introduction to network programming
- Classes and interfaces of java.net packages
- Difference between TCP and UDP
- What is Port?
- What is IP address? Types?
- What is Socket?
- Finding IP address.
- Creating URLs
- Exception classes of java.net packages
- Reading data from URL’s open stream (from local server, global server)
- Writing data through URL Connection

Module 15: Collections

- Introduction to generics & collections
- Difference between arrays and Collections
- Collection interfaces
- List, Set, Map 7 Queue Interface
- Time management over collection classes
- Memory management over collection classes
- Deleting duplicate objects from collections
- equals() method
- Iterator & Enumeration class
- for-each loop
- influence of generics in collections

Module 16: AWT, Applets and Swings

- Introduction to AWT
- AWT library classes and interfaces
- Introduction to Applets
- Creating Applets
- Applet Life cycle
- Drawing shapes and colors in Applets
- Creating components
- Layout Management, Event Listeners
- Introduction to swings
- Swings API
- Light & Heavy weight components
- Components, Menu & Sub menu creation
- Setting Listeners to components
- Project Creation using APPLETS, SWINGS, AWT and JDBC

********* PART 2: Advanced Java *********

Module 17: JDBC

- JDBC Introduction
- JDBC Architecture
- Java.Sql Package
- Connection, Statement, Result Set
- Prepared and Callable Statement
- Scrollable and Updatable Result Set
- Batch Updates
- Result Set Meta Data
- Simple Transaction Management
- Four Levels of JDBC Drivers, Their Pros & Cons
- Features of JDBC 3.0/4.0

Module 18: Servlets

- Need of server-side Programming
- Introduction to Servlets
- Servlet Life Cycle
- Javax.Servlet Package
- ServletConfig, ServletContext, ServletResponse
- Supplying initialization parameters to Servlets
- Performing database operations in Servlets
- Include And Forward Mechanisms
- Applying filters to Servlets
- Javax.Servlet.Http Package
- HttpServlet Life Cycle
- Http request methods GET vs POST

- HttpServletRequest, HttpServletResponse
- Dealing With Http Headers & Error Codes
- Session Tracking, Purpose
- Hidden form fields, Cookies
- Http Session, Url Rewriting
- Event Listeners
- Web Application Security

Module 19: JSP

- Disadvantages of Servlets
- Introduction to JSP
- JSP Life Cycle
- Creating dynamic Web content with JSP
- Scripting Elements
- Scriptlet, Declaration & Expression
- XML syntax for JSP elements
- JSP directives page, include and taglib
- JSP implicit objects & scopes
- Include And Forward Mechanism
- Using a Java bean in a JSP
- JSP Model 1 architecture
- JSP Model 2 (MVC) architecture
- Custom Tag Development, Classic Tags, Simple Tags
- Error Handling in a JSP
- JSTL
- Expression Language
- Processing XML in a JSP

********* PART 3: Spring *********

Module 20: Introduction to Spring Framework & Core Modules

- Introduction to Spring Framework and Modules
- Introduction to IOC and Spring Container
- Types of Dependency Injections
- Setter injection
- Constructor injection
- Resolving Constructor Injection Confusion
- Collection Dependency Injection
- Autowiring & Dependency Checking
- Spring bean scopes
- Factory methods Configuration
- Bean Inheritance
- Bean Life Cycle
- Aware Interfaces
- Pre and Post Processors
- Working with properties files
- Multiple Configuration files
- Null string
- Method replacement
- Lookup method injection
- Spring expression language (spel)
- Annotation-based dependency injection
- Inner beans
- Alias Names
- Lazy Initialization
- Spring Boot Application Introduction

Module 21: Spring JDBC

- Drawbacks of plain JDBC
- Templates, Jdbc Template
- Named Parameter & Simple Jdbc Template
- All the methods defined in Templates
- DAO Support classes
- Callback interfaces
- Basic & Driver manager Data source
- Examples of database access operations using spring jdbc
- Examples of accessing stored procedures using spring jdbc
- RDBMS Operation classes

Module 22: Spring ORM

- Introduction to ORM
- Integrating with Hibernate
- Spring with hibernate without Hibernate Template
- Spring with hibernate with Hibernate Template
- Hibernate Template
- Hibernate DAO Support
- Callback interfaces
- Spring with JPA and other ORM integrations

Module 23: Spring AOP

- Introduction & Need of AOP
- AOP Terminologies
- Types of Advices & Point cuts
- AspectJ Style AOP support
- Programmatic based & Declarative Based AOP

Module 24: Spring Transaction Management

- Introduction to Transaction Management
- Declaring a Spring Transaction Manager
- Configuring isolation levels
- Configuring transaction propagation
- Transactions and integration testing
- Aop 2.0 Configuration driven Transaction Management
- AspectJ annotation-based Transaction engagement

Module 25: Spring MVC

- Introduction To MVC
- Spring MVC flow & Resources
- Dispatcher Servlet
- Controller Classes
- Hadler Mappings
- Validations & View Resolvers
- Tiles in Spring MVC
- Message source
- MVC namespace (new in Spring 3.0)
- Spring Handler interceptors
- I18N
- PDF Views and Excel Views
- Spring MVC annotations
- Struts-Spring-Hibernate Integration

Module 26: Spring JEE

- Scheduling Based Applications with Spring
- Spring Webservices Integration

Module 27: Java Real Time Project 1

- Introduction to Java Realtime Project 1
- Build Java Project 1 for Realtime scenario
- Test the Examples and debug the if required

Module 28: Java Real Time Project 2

- Introduction to Java Realtime Project 2
- Build Java Project 2 for Realtime scenario
- Test the Examples and debug the if required

Module 29: Core & Advanced Java Certification Exam Prep

- Explain various Core Certification Options
- Discuss 50+ Important Python Certification Questions
- Practice Python Certification questions

Module 30: Resume Preparation, Interview and Job Assistance

- Prepare Crisp Resume as Python Developer
- Discuss common interview questions in Python
- Explain students what jobs they should target and how