

Full Stack Development using Java

Description

This Internship on java aims to provide learners both classical and modern features of the language of Java and their practical use.

Expectations and Goals

Learning programming and core Java concepts Introduction to Inheritance, Threads and Collections Deploy JDBC for connecting various applications Understand Method Overriding and Overloading Use Array and Hash Map for storing dynamic data Create Threads in Java by Implementing Runnable Interface. Work on live projects for hands-on experience.

- Live Sessions by the mentor.
- Opportunity to interact with trainer.
- After each session the recording of the session shall be provided.
- Doubt clearing sessions.
- 24/7 Support team to assist in software installation and other issues.
- Live Project implementation.
- Internship Certificate.
- Ardent Certificate contains logos of all the affiliations like Microsoft, Adobe, AutoDESK, EC-COUNCIL, MSME, NCVT, ISO 9001:2015.
- Softcopy of study materials shall be provided.

Prerequisites

Anybody can take this Training Internship to be a Java Developer.

Course Schedule

Module	Topic
Module 1	Java New Features
	Introduction to JShell
	What is JShell: An introduction to REPL (Read-Evaluate-Print-Loop)
	Why use JShell
	Starting and stopping JShell
	Starting JShell in verbose mode
	Working with snippets in JShell
	Changing definition of a variable, method or class
	Changing level of feedback
	Forward references Exception
	Tab completion for snippets
	Snippet transformation
	Shell commands
	Tab completion for commands
	Command abbreviations
	External JShell editor
	Setting classpath and module options
	JShell scripts, Startup scripts, Creating and loading scripts
	Module Declarations
What is a module: dependences and dependencies	
Module declaration syntax	
Normal and open modules	

	<p>Module and reflective access</p> <p>Module directives: requires, exports, opens, uses, provides</p> <p>Requires modifier: transitive and static</p> <p>Module file specification: module-info.java</p> <p>The primordial module: java.base</p> <p>Module practical examples</p> <p>Unnamed module</p> <p>Using jlink tool to assemble and optimize a set of modules and their dependencies into a custom runtime image</p>
Module 2	<p>Introduction to java</p> <p>Primary components of a Java program: Class, Interface, Enum and Annotation</p> <p>Writing, compiling and running a Java program from command line</p> <p>What is Java byte-code?</p> <p>JVM and JRE</p> <p>Java bytecode interpreter and JIT compiler</p> <p>How to work with Eclipse, Netbeans and IntelliJ IDE</p>
Module 3	<p>Class and Object</p> <p>What is an object: object properties and operations</p> <p>What is a class</p> <p>How does a class describe properties of objects: private fields, accessor and mutator methods</p> <p>How does a class describe operations using methods</p> <p>Method overloading</p> <p>The 'this' keyword</p>
Module 4	<p>Constructors</p> <p>What is a constructor</p> <p>Default constructor</p> <p>Constructor overloading</p> <p>Constructor chaining</p>
Module 5	<p>Static or class variables and methods</p> <p>Static variable</p> <p>Static method</p>
Module 6	<p>Some advanced class concepts</p> <p>Static and non-static field initializers</p> <p>Static and non-static initialization blocks</p> <p>Order of initializations</p> <p>Private constructors and singleton class**</p>
Module 7	<p>Nested classes**</p> <p>Static member nested classes</p> <p>Member inner classes</p> <p>Local inner classes</p> <p>Anonymous inner classes</p>
Module 8	<p>Packages in Java</p> <p>Package concept and its advantages</p> <p>How to place a class inside a package</p> <p>How to import a class</p> <p>The default access modifiers</p> <p>Compiling and running java classes in packages: concept of java classpath</p> <p>Creating jar packed libraries in java**</p> <p>Java extension mechanism**</p> <p>Creating executable jar files**</p>
Module 9	<p>Inheritance</p>

	<p>What is inheritance: java inheritance mechanism Inheriting fields and methods from superclass Adding fields and methods in subclass Upcasting, downcasting and instanceof operator Method overriding, dynamic binding and runtime polymorphism Use of 'super' keyword Constructor chaining using 'super' keyword Inheritance and access modifiers Final classes Concept of single rooted class hierarchy in java: the 'Object' class</p>
Module 10	<p>Abstract classes and interfaces Abstract method Abstract class Abstract class and inheritance Interfaces and its implementation Interfaces and multiple inheritance Interfaces and loose coupling Field declarations within an interface Marker interfaces Default implementation of methods within interface** Functional interfaces and lambda expressions**</p>
Module 11	<p>Exception handling Why do we need exception handling in java Exception handling mechanism in java using try, catch and finally Stack unwinding Difference between Exceptions and Errors 'Throwable' class Checked and unchecked exceptions Exception chaining Custom exceptions Catching multiple exceptions in java 7** Try with resources** Suppressed exceptions***</p>
Module 12	<p>Multithreading Concept of processes and threads Multithreading by extending Thread class Multithreading by implementing Runnable interface Life cycle of a thread Thread synchronization: concept of monitor, synchronized blocks and synchronized methods Inter thread communication by guarded blocks: wait, notify and notifyAll Deadlock, starvation and livelock** Lock objects** Executors**</p>
Module 13	<p>Generics Why use Generics Generic Types Raw Types Generic Methods Bounded Type parameters Generics, Inheritance and Subtypes Type inference Wildcards</p>

	Type Erasure Restrictions on generics
Module 14	Java collection framework Introduction to Java collection framework Core collection interfaces and their implementations: Collection, Set, List, Queue, Deque, Map, SortedSet and SortedMap Aggregate operations: Reduction and Parallelism** Algorithms: Sorting, Shuffling, Routine data manipulation, Searching, Composition and finding extreme value
Module 15	Basic I/O Concept of Input and Output in Java Byte Streams and Character Streams Buffered Streams Scanning and Formatting Command Line I/O Data and Object Streams File I/O: Nio.2**
Module 16	Annotations Annotation Basics Declaring an Annotation Type Predefined Annotation Types Type Annotations and Pluggable Type Systems** Repeating Annotations**
Module 17	Sub Language Commands Data Definition Language (DDL) Data Retrieval Language (DRL) Data Manipulation Language (DML) Transaction Control Language (TCL) Database Security and Privileges (DCL) Oracle Pre Defined Datatypes DDL Commands Create, Alter (add, modify, rename, drop)Columns, Rename, truncate, drop DML-Insert, update, delete DQL-SELECT Statements using WHERE clause Comparison and Conditional Operators Arithmetic and Logical Operators Set Operators (UNION, UNION ALL, INTERSECT, MINUS) Special Operators – IN (NOT IN), BETWEEN (NOT BETWEEN), LIKE (NOT LIKE), IS NULL (IS NOT NULL) Working with DML, DRL Commands
Module 18	JDBC Introduction to JDBC JDBC architecture java.sql Package Connection, Statement, ResultSet Prepared Statement Callable Statement Scrollable and Updatable ResultSet Batch Updates ResultSetMetaData

	Simple Transaction Management Four Levels of JDBC drivers, their pros & cons Features of JDBC 3.0
Module 19	Java Swing Desktop application Java applets Java Swing Swing controls Design GUI
Module 20	Project work and documentation