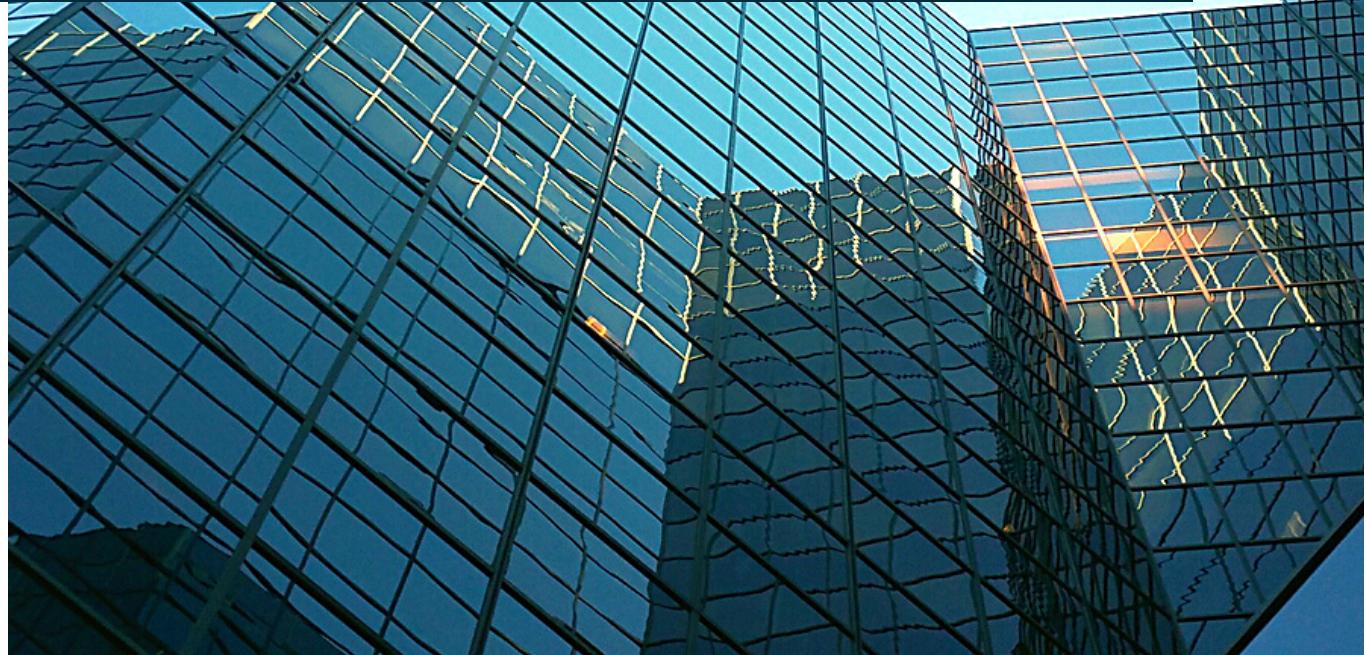




JAVA Full stack Content

A better learning future starts here



GET TRAINED AND GET EMPLOYED!

About 4Achievers

4achievers is one of the leading organizations in the education industry. We specialized in the quality driven training. We focus on emerging technologies and processes that are transforming the digital world, at a fraction of the cost and time as traditional approaches.

We believe in hard work and we know that learning is a key to success hence we help our each and every student to achieve their future goals with continuous learning and smart work.

We are providing one on one counselling to students with the trainer so that if any student wants to discuss their doubts with the trainer then they are most welcomed to do so.

On our website, we have already provided the students with all kind of digital counselling so that the student can first get satisfied by the digital learning and then they can communicate with the respective trainer on their doubts.

We want to build a family where each student is treated as important as a family member.

CORE JAVA

Course Content

J2SE:[Core Java]:

- **Introduction:**
 - Java History
 - Differences between java and others
 - Java Features
 - Java Naming Conventions
 - Java Programming Format
- **First Java Application Development:**
 - Java Installation
 - Editor
 - Java Application and Java File Saving.
 - Compile Java File
 - Execute Java Applications.
- **Language Fundamentals:**
 - Tokens
 - Operators
 - Identifiers
 - Literals
 - Key Words /Reserved Words
 - Data Types and Type casting
 - Java Statements
 - Arrays
- **OOPS**
 - A.
 - Types of Programming Languages
 - Unstructured Programming Languages
 - Structured Programming Languages
 - Object Oriented Programming Languages
 - Aspect Oriented Programming Languages

- B
 - Class
 - Class
 - Object
 - Encapsulation
 - Abstraction
 - Inheritance
 - Abstraction
 - Polymorphism
 - Message Passing
- C
 - Object Based PL VS Objec
 - Oriented PL
 - Class syntax
 - Method Syntax
 - Var-arg method.
 - Accessor Methods VS Mutator
 - Methods
 - Syntax to create an object
 - Immutable Objects VS Mutable
 - Objects
 - Object Vs Instance
 - Constructors
 - Default Con
 - Default Con.
 - User defined con.
 - 0-arg-con.
 - param-con.
 - Instance Context
 - Instance
 - variable
 - Instance block.
 - Instance method
 - This keywords
 - To refer current class variable
 - To refer current class methods
 - To refer current class blocks
 - To return current class objects

- Static keyword
 - Static variable
 - Static block
 - Static method
 - Static import
- Main () method
 - Public static void main (String [] args)
 - Why public?
 - Why static?
 - Why void?
 - Why main?
 - Why String [] as parameter?
 - Is it possible to overload main (-) method?
 - Is it possible to
 - override main (--) method?
- Is it possible to provide more than one main
- (--) method with in a single java appl?
- Is it possible to execute any java application without using main method?
- Factory Method
 - Singleton classes and Doubleton classes
 - Final Keyword
 - Final variable
 - Final class
 - Final method
 - Enum keyword
 - Relationships in JAVA
 - IS-AVs HAS-A Vs USE-A
 - Assiciation in Java
 - one-one
 - many-one
 - one-many
 - many-many
 - Inheritance and Types of inheritances
 - Single
 - Hierarchical

- Hybrid.
- Multilevel
- Static flow in inheritance
- Instance flow in inheritance
- Super keyword
- Class level type casting
- Poly Morphism
- Static PM
- Dynamic PM
- Method overloading
- Method overriding
- Abstract Methods Vs Concrete Methods
- Abstract class Vs concrete Class
- Class Vs Abstract class Vs interface
- Instance of" operator
- What is Adapter class?
- What is marker interface?
- Object Cloning
 - Shallow Cloning
 - Deep Cloning
- JAVA8 features in interfaces
- **Inner classes:**
 - Member
 - Inner class
 - Method local Inner class
 - Static Inner class
 - Anonymous Inner class
- **Wrapper classes:**
 - Byte,
 - Short
 - Integer
 - Long
 - Float
 - Double
 - Boolean
 - Character

- Packages:

- What is a package?
- Adv. of packages
 - Modularity
 - Reusability
 - Abstraction
 - Sharability
 - Security
- Types of packages
 - Predefined packages
 - User defined packages
- Jar files preparation
- Executable Jar files
- Batch files preparation

- String manipulations:

- String 3
- String Buffer 3
- String Builder
- String to kenizer

- Exception Handling:

- Error VS Exception
- Exception Def.
- Types of Exceptions
- Predefined Exceptions 2. User defined Exceptions
- Checked Exception VS Unchecked Exception
- Pure Checked Exceptions
- Partially Checked Exceptions
- Throw Vs throws
- try-catch-finally
- Custom Exceptions
- Java7 Features in Exception Handling
 - Automatic Resource management
 - Multi catch block.

- **Multi-Threading:**

- Process Vs Processor Vs Procedure
- Single Processing Mech. Vs Multi Processing Mech.
- Single Thread model And Multi Thread Model
- Thread Design
 - Extending Thread class
 - Implementing Runnable interface.
- Thread lifecycle
- New/Born
- Runnable
- Running
- Blocked
- Dead
- Thread class library
 - Sleep ()
 - Join ()
 - Yield ()
 - Stop ()
- Daemon Thread
- Synchronization
- Inter Thread communication
 - Wait ()
 - Notify ()
 - Notify All()

- **IOStreams:**

- What is stream?
- Types of Streams?
 - Byte-oriented Stream
 - Input Streams
 - Output Streams
 - Character-Oriented Streams
 - Reader
 - 2. Writer
- File Input Stream Vs File Output Stream
- File Reader Vs File Writer
- File Vs Random Access File
- Serialization vs Deserialization
- Externalization

- **Networking:**
 - Standalone Appl. Vs Distributed Appl.
 - Client-Server Arch.
 - Socket Vs Server Socket
 - Network Appl. Arch.
 - Socket Programming.
- **AWT:**
 - Text Field, Text Area, Button, Label, Check Box, List.
 - Swing:
 - J Text Field, J Password Field, J Check Box, J Radio Button, J Color Chooser.
 - Event Delegation Model
- **I18N:**
 - Number Format
 - Date Format
 - Resource Bundle
- **Reflection API:**
 - Class
 - Field
 - Method
 - Constructor
- **Annotations:**
 - What is Annotation?
 - Adv of annotations
 - Comments Vs Annotations
 - Types Of annotations
 - Built-in Annotations
 - @Override @Inherited @Deprecated @Target
@Suppress Warnings @Documented @Retention
 - User Defined Annotations

- **Remote Method Invocation[RMI]:**
 - Introduction
 - RMI Architecture
 - Steps to Design RMI Application
 - Parameters in Remote methods
- **Regular Expressions:**
 - Introduction
 - Pattern
 - Character
 - Quantifiers
- **Garbage Collection:**
 - Introduction
 - Approaches to make an object for GC
 - Methods for requesting JVM to run GC
 - Finalization
- **JVM Arch.**
 - Class Loading Sub System
 - Memory Management System
 - Execution Engine
 - Java Native Interface
 - Java Native library
- **Generics:**
 - Introduction
 - Generic Classes
 - Generic Methods & Wild Card Character.
 - Inter Communication with Non-Generic Code
- **Java 8 Features:**
 - Lambda Expressions
 - Functional Interfaces
 - Default Methods in Interfaces
 - Static Methods in Interfaces
 - Predicate
 - Function
 - Consumer
 - Method Reference & Constructor reference Double Colon (::: operator.)

- Stream API
- Date & Time API (Joda API.)
- **JAVA 9 NEW FEATURES:**
 - The Java Shell (RPEL.)
 - The Java Platform Module System(JPMS.)
 - JLINK(JAVA LINKER.)
 - Process API Updates
 - Private Method in Interfaces.
 - Factory Methods for Collections.
 - Enhancements to Java 8 Stream API
 - Try With Resources Enhancements.
 - Diamond Operator
 - SafeVargs Annotation
 - HTTP/2 Client
 - G1 Garbage Collector
- **Java 10 Updations:**
 - Local-Variable Type Inference
 - Consolidate the JDK Forest into a Single Repository
 - Garbage-Collector Interface
 - Parallel Full GC for G1
 - Application Class-Data Sharing
 - Thread-Local Handshakes
 - Remove the Native-Header Generation Tool (javah)
 - Additional Unicode Language-Tag Extensions
 - Heap Allocation on Alternative Memory Devices
 - Experimental Java-Based JIT Compiler
 - Root Certificates
 - Time-Based Release Versioning
- **Java 11 Version Updations:**
 - Running Java File with single command
 - New utility methods in String class
 - Local-Variable Syntax for Lambda Parameters
 - Nested Based Access Control 5. HTTP Client
 - Reading/Writing Strings to and from the Files
 - Flight Recorder

- **Java 12 Features:**

- Switch Expressions
- File mismatch() Method
- Compact Number Formatting
- Teeing Collectors in Stream API
- Java Strings New Methods – indent(), transform(), describeConstable(), and resolveConstantDesc().
- JVM Constants API
- Pattern Matching for instanceof
- Raw String Literals is Removed From JDK 12.

- **Java 13 Updations:**

- Text Blocks
- New Methods in String Class for Text Blocks
- Switch Expressions Enhancements
- Reimplement the Legacy Socket API
- Dynamic CDS Archive
- ZGC: Uncommit Unused Memory
- FileSystems.new
- FileSystem() Method

- **JAVA 14 Updations:**

- Developers Required Features:
- Switch Expressions (Standard)
- Pattern Matching for instanceof (Preview)
- Helpful NullPointerExceptions
- Records (Preview)
- Text Blocks (Second Preview)

- **Collection Framework:**
 - Collection Arch.
 - List and its implementations
 - ArrayList
 - Vector
 - Linked List
 - Stack
 - Set and its implementations
 - HashSet
 - Linked HashSet
 - TreeSet
 - Map and its implementations
 - HashMap
 - Hash table
 - Properties
 - TreeSet
 - Queue and its implementations
 - PriorityQueue
 - Blocking Queue
 - Priority Blocking Queue
 - Linked Blocking Queue
 - Iterators
 - Iterator
 - List Iterator
 - Enumeration

JDBC:

- **Storage Areas**
 - Temporary Storage Areas
 - Permanent Storage Areas
- **Query Processing System**
 - Query Tokenization
 - Query Processing
 - Query Optimization
 - Query Execution

- **Driver and Driver Types**

- Type 1 Driver
- Type 2 Driver
- Type 3 Driver
- Type 4 Driver

- **Steps to design JDBC Applications**

- Load and register the Driver.
- Establish the connection between Java Application.
- Prepare either Statement or prepared Statement
- Write and execute SQL Queries.
- Close the connection.

- **Result Set and Result Set Types**

- Read only Result Set
- Updatable Result Set
- Forward only Result Set
- Scrollable Result Sets Scroll Sensitive Result Set
 - Scrollable Result Sets Scroll Sensitive Result Set
 - Scroll Insensitive Result Set

- **Prepared Statement**

- Prepared Statement with insert sql query
- Prepared Statement with update sql query
- Prepared Statement with update sql query
- Prepared Statement with update sql query
- Prepared Statement with select sql query
- Prepared Statement with Dates Handling
- Prepared Statement with Batch Updations

- **Callable Statement**

- Callable Statement with procedure
- Callable Statement with function
- Callable Statement with CURSOR Type Procedure
- Callable Statement with CURSOR type function

- **Transaction Management**

- Atomicity
- Consistency
- Isolation
- Durability
- Savepoint
- Batch Updations
- Connection Pooling
- BLOB and CLOB
- RowSets

SERVLETS:

- **Introduction**
 - Standalone Applications
 - CUI Applications
 - GUI Applications
- **Enterprise Applications**
 - Web Applications
 - Distributed Applications
- **Client-Server Arch**
 - Client
 - Protocol
 - Server
- **Servlets Design**
 - Servlet interface
 - Generic Servlet
 - Http Servlet
- **Servlet Lifecycle**
- **User Interface**
 - Static Form Generation
 - Dynamic Form Generation
- **Servlet Config**
- **Servlet Context**
 - Servlet Communication
 - Browser-servlet
 - Send Redirect Mechanism
 - Web-component
 - Include Mechanism
 - Forward mechanism
- **Session Tracking Mechanisms**
 - HttpSession Session Tracking Mechanism
 - Cookies Session Tracking Mechanism
 - URL-Rewriting Session Tracking Mechanism
- **Hidden Form Fields Session Tracking Mechanism**

- **Servlets Filters**
- **Servlets Wrappers**
 - ServletRequest Wrapper
 - HttpServletRequest Wrapper
 - ServletResponse Wrapper
 - HttpServletResponse Wrapper
- **Servlets Listeners**
 - Request Listeners
 - Context Listeners
 - Session Listeners
- **Web Security**
 - Programmatic Approach
 - Declarative Approach

JAVA SERVER PAGES:

- **Introduction**
- **JSP Deployment**
- **JSP Life Cycle**
- **JSP Elements**
 - JSP Directives
 - Scripting Elements
 - JSP Actions
- **JSP Directives**
 - Page Directive
 - Include Directive
 - Taglib Directive
- **JSP Scripting Elements**
 - Declarations
 - Scriptlets
 - Expressions
- **JSP implicit objects**
 - Out
 - Request
 - Response
 - Config
 - Application

- Out
- Request
- Response
- Config
- Application
- Session
- Exception
- Page
- Context

- **JSP Scopes**

- Page Scope
- Request Scope
- Application Scope
- Session Scope

- **JSP Standard Actions**

- <jsp:useBean>
- <jsp:setProperty>
- <jsp:getProperty>
- <jsp:include>
- <jsp:forward>
- <jsp:param>
- <jsp:plugIn>
- <jsp:fallback>
- <jsp:params>
- <jsp:declaration>
- <jsp:scriptlet>
- <jsp:expression>

- **JSP Custom Actions**

- Tag
- Iteration Tag
- Body Tags
- TagSupport
- Body Tag Support
- Simple Tag
- Simple Tag Support
- Nested tags

- **JSTL**

- Core Tags
- XML Tags
- Internationalization or I18N Tags (Formatted tags)
- SQ Tags
- Functions tags
- **Expression Language**
 - EL operators
 - EL implicit objects.
 - EL functions.

SPRING Course Content

- **Introduction:**
- **Enterprise Appl**
- **Enterprise Application Layers**
 - Presentation Layer
 - Business Layer
 - Data Access Layer
- **System Architectures**
 - 1-Tier Arch.
 - 2-Tier Arch.
 - n-Tier Arch
- **Types of Enterprise Applications.**
 - Web Applications
 - Distributed Applications
- **Modeled Arch.**
 - Model-I Arch.
 - Model-II Arch.
- **MVC**
- **Requirement to user Frameworks**
- **Types of Frameworks**
 - Web Frameworks
 - Application Frameworks
- **Differences between Spring and Struts, JSF**
- **Spring History**
- **Spring Modules**

- Spring1.x Modules
- Spring2.x Modules
- Spring3.x Modules
- Spring4.x Modules
- Spring5.x Modules

- **Steps To Prepare Spring Application [Core Module Application]:**

- Download Spring Framework from Internet.
- Provide Spring Setup in Eclipse IDE
- Prepare Bean Class
- Prepare Bean Configuration File
- Prepare Test / Client Appl.

- **Core Module**

- Introduction
- IOC Containers
 - BeanFactory
 - XmlBeanFactory
 - Resources
 - ByteArray Resource
 - FileSystem Resource
 - ClassPath Resource
 - InputStream Resource
 - Url Resource
 - Servlet Context Resource
 - Portlet Context Resource
 - ApplicationContext
 - ClassPathXmlApplicationContext
 - FileSystemXmlApplicationContext
 - WebXmlApplicationContext
 - Beans in Spring Framework
 - Beans Definition
 - Beans Configuration
 - XML Based Configuration
 - Annotation Based Configuration
 - Java Based Configuration

- Bean Scopes
 - singleton Scope
 - prototype Scope
 - request Scope
 - session Scope
 - globalSession Scope
 - application Scope
 - webSocket scope
 - Custom Scopes in
 - Spring Framework.
- Bean Lifecycle
 - Bean Loading
 - Bean Instantiation
 - By Constructor
 - By Static Factory Method
 - By Instance Factory Method
 - Bean Initialization and Destruction
 - By Custom initialization and destruction methods.
 - By InitializingBean and Disposable Bean callback interfaces.
 - By @PostConstruct and @Predestroy Annotations
- Beans Inheritance
- Nested Beans
- BeanPostProcessor
- Inversion Of Control[IOC]
 - Dependency Lookup
 - Dependency Pull
 - Contextualized Dependency Lookup
 - Dependency Injection
 - Constructor Dependency Injection
 - Setter Method Dependency Injection
 - Different Types of Elements Injection
 - User defined data types elements injection.
 - List types injection
 - Set types injection
 - Map Types Injection
 - Properties types Injection

- Circular Dependency Injection
- Name Spaces
 - P-Name space
 - C-Name Space
- Beans Autowiring or Beans Collaboration
- Autowiring and its Modes
- no
 - ByName
 - ByType
 - Constructor
- Annotation Based Wiring
- Autodiscovery or Stereo Types
- Java based Autowiring[Java Based Configuration]
- Method Injection
 - Looku Method Injection
 - Arbitrary Method Replacement
- Event Handling
 - ContextRefreshedEvent
 - ContextStartedEvent
 - ContextStoppedEvent
 - ContextClosedEvent
 - RequestHandledEvent
 - Custom Events In Spring Framework
- Bean Validations in Spring Framework
- Internationalization in Spring Framework
- Bean Manipulations and Bean Wrappers
- Property Editors
 - ByteArrayPropertyEditor
 - ClassEditor
 - CustomBooleanEditor
 - CustomCollectionEditor
 - CustomNumberEditor
 - FileEditor
 - InputStreamEditor
 - LocaleEditor
 - PatternEditor
 - PropertiesEditor

- StringTrimmerEditor
- URLEditor
- Custom Property Editors [User defined]
- Profiling
- Spring Expression Language[SpEL]
 - SpEL Expressions
 - SpEL Operators
 - SpEL Variables
 - SpEL Method Invocations
 - SpEL Collections

Spring JDBC/DAO Module:

- **Introduction**
- **DAO Definition**
- **Advantages of DAOs**
- **Drawbacks with DAOs**
- **Guidelines to prepare DAOs**
- **Pain JDBC Vs Spring JDBC**
- **Jdbc Template**
- **NamedParameterJdbcTemplate**
 - Parameter values through Map
 - Parameter Values through SqlParameterSource
 - MapSqlParameterSource
 - Bean PropertySqlParameterSource
- **SimpleJdbc Template**
- **DAO Support Classes**
 - JdbcDaoSupport
 - NamedParameterJdbcDaoSupport
 - SimpleJdbcDaoSupport
- **Spring Batch Updations or Batch Processing**
- **Stored Procedure and Functions in Spring JDBC**
 - Procedures and Functions without CURSOR Types
 - Procedures and Functions with CURSOR Types
- **Blob and Clob processing in Spring JDBC**
- **Abstract Lob Creating Prepared Statement Call back**
- **Abstract Lob Streaming Result Set Extractor**

- Lob Creator
- Lob Holder
- Connection Pooling in Spring JDBC
 - Default Connection Pooling Mech.
 - Third Party Connection Pooling Mechanisms
 - Apache DBCP
 - C3P0
 - Proxool
 - Application Servers provided Connection Pooling Mechanism
 - Weblogic12c provided Connection Pooling Mechanism.

Spring ORM

- **Introduction**
 - Hibernate Integration with Spring
 - Hibernate Introduction
 - Hibernate Application Development
 - Spring with Hibernate Integration.
 - JPA Integration with Spring
 - JPA Introduction.
 - JPA Application development
 - Spring with JPA Integration.
 - iBatis integration with Spring
 - iBatis Introduction.
 - iBatis Application Development.
- **Spring with iBatis Integration.**

Aspect Oriented Programming [AOP]

- **Introduction**

- **AOP Terminology**

- Aspect
- Advice
- JoinPoint
- Pointcut
- Introduction
- Target
- Proxy
- Weaving
- Advisor

- **Types of AOPs**
 - Proxy Based AOP
 - Declarative Based AOP
 - Annotation Based AOP
- **Advices**
 - Before Advice
 - After Advice
 - After-Returning Advice
 - Around Advice
 - After-Throwing Advice
- **Pointcuts**
 - Static Pointcut
 - Dynamic Pointcut.

Spring Transactions

- Introduction
- Transaction Attributes
- Isolation Levels
- Programmatic Based Transactions
- Declarative Based Transactions.
- Annotation Based Transactions

Spring web MVC Module

- **Introduction**
- **Spring MVC Flow**
- **Controllers**
 - Abstract Controller
 - ParameterizableViewController
 - MultiActionController
 - Command Controllers
 - AbstractCommandController
 - AbstractFormController
 - SimpleFormController
 - AbstractWizardFormController
 - Handler Mappings
 - BeanNameUrlHandlerMapping
 - SimpleUrlHandlerMapping

- **HandlerInterceptor**
- **ViewResolvers**
 - AbstractCachingViewResolver
 - XmlViewResolver
 - ResourceBundleViewResolver
 - UrlBasedViewResolver
 - InternalResourceViewResolver
 - VelocityViewResolver / FreeMarkerViewResolver
- **Spring Exception Handling**
- **File Uploading and File Downloading**
- **Internationalization**
- **Spring MVC with Tiles**

Spring Web :

- Introduction
- Spring Integration with Struts.
- Spring Integration with JSF.

Spring Security

- Spring Security Introduction
- Spring Security Features
- Spring Security XML Based Example
- Spring Security Java Based Example

Spring Boot

- Introduction
- Spring Boot Features/ Advantages.
 - Spring Boot Starters.
 - Spring boot Auto configurations
 - Spring Boot Embedded Containers
 - Spring Boot Actuators
 - Spring boot Test
- Spring boot Core Applications [Core Module].
- Spring boot JDBC Applications [JDBC Module].
- Spring boot Hibernate Application [ORM Module]
- Spring Boot JPA Application [ORM module]
- Spring Boot Data-JPA Application [ORM Module]

- Spring Boot Transaction Application Transaction module,
- Spring boot Web MVC Application[Web MVC Module
- Spring boot security
- Springboot Microservices
- SpringBoot Cloud

- **Introduction**
 - Enterprise
 - Enterprise Application
 - Enterprise Application Layer
 - User Interface Layer
 - Business Processing Layer
 - Data Storage and Access Layer
 - Data Persistency
 - Data Persistency through Serialization and Deserialization
 - Data Persistency through JDBC
 - Data Persistency through ORM
 - Paradigm Mismatches
 - Granularity Mismatch
 - Sub Types Mismatch
 - Associations Mismatch
 - Identity M3ismatch
 - EJBs Vs Hibernate
 - JPA Vs Hibernate
- **Hibernate History**
- **Hibernate Features**
- **Hibernate Arch.**
- **Steps to Prepare Hibernate Application**
 - Persistence Class / POJO class
 - Mapping File
 - Hibernate Configuration File
 - Client Application
- **Hibernate Applications**

- Hibernate Application with Main Class as Client.
- Hibernate Application with GUI Application as Client.
- Hibernate Application with Servlet as Client.
- Hibernate Application with JSP Page as Client.
- Hibernate Application with Struts Application as Client.
- Hibernate Application with MYSQL DB
- Hibernate Application with Multiple DBs [Oracle DB and MySQL DB]
- Hibernate Basic Annotations [Without Mapping File]
- Hibernate Application without Configuration File
- Hibernate Application with Composite Keys.

- **Hibernate Persistence Object Lifecycle**

- Transient State
- Persistent State
- Detached State
- Removed State
- Hibernate Tools
 - Schema Export
 - Schema Update
 - Code Generation
- Primary Key Generation Algorithms [XMI and Annotations]
 - Assign
 - Increment
 - Sequence
 - Identity
 - Hilo
 - Seq-Hilo
 - Native
 - UUID
 - Foreign
 - G UID
 - Select
- Transaction Management
 - ACID Properties
 - Automicity
 - Consistency
 - Isolation
 - Durability

- Transaction Management in JDBC
 - Automicity
- Achievement in JDBC
 - Isolation Problems
- Transaction Management in Hibernate

Hibernate Query Language [HQL]

- **HQL Elements**

- Clause
- 'From' Clause
 - 'Select' Clause
 - 'Where' Clause
 - 'Order by' Clause
 - 'Group by' Clause
 - 'Having' Clause

- **Aggregate Functions**

- count(-)
- sum(-)
- min(-)
- max(-)5avg(-)

- **Generic Expressions**

- Arithmetic Operators in Generic Expressions
- Comparision Operations in Generic Expressions
- Scalar Functions in Generic Expressions
 - In
 - Between
 - Like
 - is null
 - is not null
- Associations and Joins
- Parameters
 - Positional parameters
 - Names Parameters
- Subqueries
 - Pagination
 - HQL with Updations

- Native SQL
 - Scalar SQL Queries
 - Stored Procedures and Functions
- Criteria API
- Hibernate Filters
- Hibernate Mappings
 - Basic 'OR' Mapping
 - Component Mapping
 - Inheritance Mapping
 - Table per Class Hierarchy
 - Table per Sub-Class
 - Table per Concrete Class
 - Associations Mapping
 - One-To-One Association
 - One-To-Many Association
 - Many-To-One Association
 - Many-To-Many Association
- Connection Pooling
 - Inbuilt Connection Pooling Support in Hibernate.
 - Third Party Connection Pooling Mechanisms C3P0, Proxool, DBCP.....
 - Connection Pooling through Weblogic Server JNDI.
- Cache Mechanisms
 - I level Cache
 - II Level Cache

UI Technologies Course

Course Content

- **HTML**
- **CSS**
- **JAVA SCRIPT**
- **JQUERY**
- **BOOTSRAP**
- **DOM**
- **Angular JS-8**
 - Introduction
 - Directives
 - one way data binding
 - two way data binding
 - Life-cycle-hooks
 - Components
 - Modules
 - Template-driven-forms
 - Reactive-Forms
 - Pipes
 - Services
 - Routing
 - Crud operation using HttpService
- **DATABASE**
 - Oracle
 - MySQL
 - Mongo DB
- **SERVER**
 - Tomcat
 - Weblogic
 - JBOSS and Wildfly
 - Glassfish

- IDE's
 - Eclipse
 - Spring Tool Suite (STS)
 - IntelliJ Idea
 - Netbeans
- Real Time Tools:
 - Maven
 - Log4j

C-58, Ground Floor, Sector 2,
Near Priya Gold Building, Noida, 201301

Certificate



Review

Sajid Ahmad



Review 5



4achiever's training is one of the best training, I have ever taken. I am so pleased to say this that the AWS training that I have taken is amazing. I consulted them for AWS solution architect associate training but they helped me to understand that alone doing this certification or training could not be of any advantage. Mam helped me to understand that AWS is just only a cloud infrastructure if we really need to excel as a cloud engineer then first, we need to improve our networking then we need to make our hand on strong on Linux operating system and then we should move on with AWS part. It was so great learning here; at every moment of time, I was helped by the trainer just like a family. I am happy to say that I am placed in a very good company as a Linux administrator. Thanks a lot, 4achievers team

Swaranjali Yadav



Review 4.8



I am doing AWS solutions architect associate training, the training was started from very basic level. With one of the most experienced trainer Megha Kaushik ma'am, This is a very good effort of her that providing online classes on zoom app. She is friendly, kind, knowledgeable and great fun to talk with. He knows what is important for learners and also what we need to do to improve quickly.

Shubham Sharma



Review 5



I joined the 4achievers few months ago, first time I heard about this training institute from social media. First they provided the free demo lecture. They have most experienced staff for training, quality of training trainer will explain each and every concepts from basic level. So it will be easier to grab an every concept, it will be best for us. All these things 4achievers Will provide in very cost effective, with placement facilities.

Review

Naman Deep Pandey



Review 4.8



I joined the 4achievers few months ago, first time I heard about this training institute from social media. First, they provided the free demo lecture. They have most experienced staff for training, quality of training trainer will explain each and every concept from basic level. So, it will be easier to grab every concept, it will be best for us. I would definitely recommend anyone this institute for better job opportunities and best job placement in AWS field.

Sourabh Yaduvanshi



Review 4.8



I would say that I am very pleased to be the part of the institute, I wanted to learn RHCSCA as there are better job opportunities in the market after learning Linux. I was right to join the institute and with the help of trainer I got selected in a good company after doing RHCSCA, ansible and AWS in a combo pack from the institute. The institute is a hope for everyone who want to get placed at a high level with good salary package.

Aditi Pundir



Review 4.8



I hope after the training the 4achievers will provide the best placement facilities to us. And stay connected with us for future concern.

Review

Bharat



Review 4.8



I am currently taking AWS training from 4achievers, When I joined the institute, I had no idea of how to proceed in this new career but everyone in 4achievers were so motivating that I finally choose to attend the training. The trainer is very perfect in her work, she first taught us how to get excel with Linux and then she take on to AWS and rest she also helps us in moving with fully automation part . she is perfect with her teaching style. I loved 4achievers and I would recommend this to everyone undoubtedly. I have learned a lot and I am highly hopeful of achieving my dream job very soon.

Aboli



Review 4.8



I am from Hyderabad, Though Hyderabad is already a high-tech place and best job and training opportunities are already present here but if I need to compare the training quality then I would say that 4achievers institute is perfect in all aspect. The practical training that I gained here for Learning Linux automation and AWS was amazing. Definitely a wonderful experience. They have a strong HR connection too. A perfect place for starting new career. They just not only teach us; they get us job ready and that is why 4achievers is different from other institutes and A best place to start a new life journey.

Maneesha Dwivedi



Review 4.8



The 4achievers is one of the institute that providing best training With most knowledgeable, experienced staff & providing placement facilities as well.

Review

Somnath Kale



Review 4.8



I am from Delhi; I can only say that the institute has given me the ray of hope. I was lost that how should I take my career forward. I am from mechanical branch and because of this COVID-19 period I was struggling so hard, then when I contacted the institute, they asked me to prepare for AWS platform. In the starting phase definitely, things were very difficult as it was a complete change of technology but then slowly and slowly with the help of trainers, I could easily grab the topics practically as well as theoretically and now I am so happy that I joined 4achievers and it is one of the best institutes that has fulfilled my dream of becoming a system administrator. A big thanks to 4achievers team.

Parikshit

4.6 lac package



Review 4.8



I joined 4achievers because I wanted to improve my job prospect, I am already working as a software tester but I wanted to move to Devops career so I joined the training for learning dockers, ansible, Kubernetes and Jenkins and I can say that this institute is perfect in all its form. They not only teach us but also help us to gain practical implementation by working on real time projects in their own company. A wonderful and unforgettable experience of learning at 4achievers.

FASTEST GROWING NETWORK IN CORPORATE INDUSTRY

