



BECOME A MASTER IN JAVA PROGRAMMING

Your Path to Mastery Begins Here!



JAVA PROGRAMMING

Course Mission

SevenMentor Institute leads the way in providing cutting-edge IT Training and Skill Development across India. We have strived to establish an ideal learning atmosphere at all our training centers. We prepare our students to become dependable future professionals. Our institute aspires to promote universal access to learning for students. To achieve our mission of promoting better learning we invite all students to enroll in our Java Programming Course.

Join us today to make a fulfilling career in the Development sector.

JAVA PROGRAMMING STATISTICS:



INDUSTRY INSIGHTS

85%

Website have some form of
Java Programming based
Applications

34%

Year on Year Growth For All
Java Programming Jobs

Learn Java Programming and Be in Demand Always!

In today's rapidly evolving Tech Industry, being a skilled Java Programmer is a valuable asset. There is an unprecedented requirement of all rounders and dynamic developers in the website building domain. SevenMentor's comprehensive Java Course is designed to equip you with the knowledge and practical skills to handle every aspect of Java Programming.

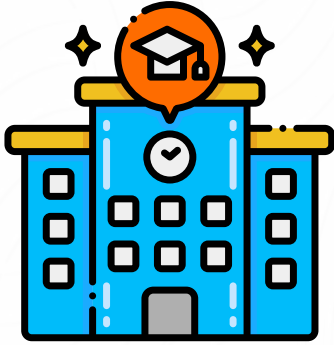
Become an Omnipresent **JAVA PROGRAMMING!**



Java Programming is like being the architect of the digital world. It's the art of seamlessly connecting the visible, user-facing Front-End with the hidden, behind-the-scenes back-end of web sites and applications. A Java Programmer possesses a unique set of skills, combining the creative flair of a front-end developer with the problem-solving prowess of a back-end developer. Whether you dream of working with innovative startups, established tech giants, or even launching your own projects, the possibilities are endless.

Start Learning to Start Applying:

Gain practical Java Programming experiences through well designed courses, latest tools and excellent teachers.



Experienced Faculty



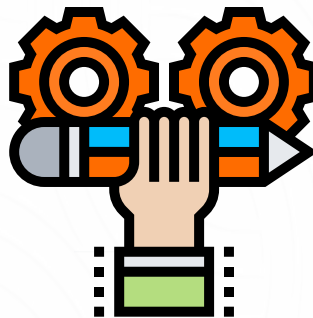
Flexible Scheduling



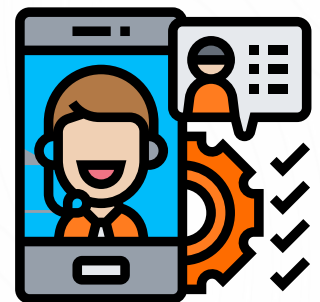
Hands-On Learning



Mock Interview Sessions



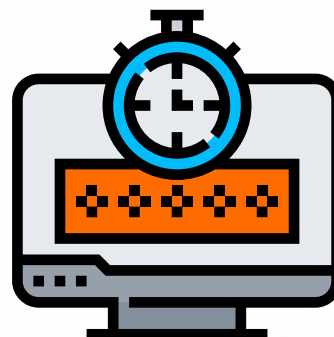
Real-World Projects



Career Support



Comprehensive Curriculum



Lifetime Access

Core Java

Duration: 2 months

Object Oriented Programming Concepts

- Object
- Characteristics of an Object
- State
- Behavior
- Identity
- Responsibility
- Major Pillars
- Abstraction
- Encapsulation
- 'IS A' relationship – Inheritance
- 'HAS A' relationship – Containment
- Polymorphism

Introduction to Java

- Features of Java
- Java Compiler
- Java Runtime Environment
- Bytecode Verifier
- Class Loader
- JIT Compiler
- JDK, JRE, JVM

Java Language Fundamentals

- | | |
|--|---|
| <ul style="list-style-type: none">• Classes and Object• Class Syntax• Primitive Data Types• Implicit and Explicit Conversion• Access specifiers• Constructor• Creating object• Memory Allocation• Garbage Collection• Method Overloading• Parameterized Constructor• this keyword | <ul style="list-style-type: none">• static keyword – variable & method• main method• Instance init block & Static init block• Array – 1D, 2D• Bytecode Verifier• Class Loader• JIT Compiler• JDK, JRE, JVM |
|--|---|

Java Language Fundamentals

- Classes and Object
- Class Syntax
- Primitive Data Types
- Implicit and Explicit Conversion
- Access specifiers
- Constructor
- Creating object
- Memory Allocation
- Garbage Collection
- Method Overloading
- Parameterized Constructor
- this keyword
- static keyword – variable & method
- main method
- Instance init block & Static init block
- Array – 1D, 2D
- Enhanced for loop - forEach loop
- Variable Argument
- Package
- Importing from packages
- static import

Containment, Inheritance and Polymorphism

- Implementing 'HAS A' relationship
- Container Object and Contained Object
- Implementing 'IS A' relationship
- extends keyword to achieve
- Inheritance
- Super class and Sub class
- super keyword
- protected keyword
- Polymorphism
- Method Overriding
- Super class reference to create Sub class object
- Reference Type Casting
- Static Type and Dynamic Type of reference
- Covariant return type
- java.lang.Object class and it's methods – toString ()

Abstract Class and Interface

- abstract keyword
- Abstract class vs Concrete class
- Abstract method vs Concrete method
- final keyword – variable, method, class
- final method vs abstract method
- final class vs abstract class
- Interface
- Role based inheritance
- Multiple Inheritance
- implements keyword
- Implementing interface in a class
- Interface extends interfaces

Annotations and Enum

- Annotation
- Creating user defined annotation
- Use of annotation at different levels
- Meta-annotation
- java.lang.annotation.Annotation interface
- Built-in Annotations - @Override, @SuppressWarnings, etc
- Enum
- Creating user-defined Enum
- Enum Constants
- Java.lang.Enum class

Utility Classes (java.lang) & Inner Classes

- Wrapper classes
- Autoboxing & Autounboxing
- String, StringBuffer, StringBuilder
- String constant pool
- Inner classes
- Simple Inner class
- Static nested class
- Method Local Inner class

Functional-Style Programming

- Functional Interface
- Lambda Expression
- @FunctionalInterface
- Method References
- Built-in Functional interface
- java.util.function package
- Predicate, Consumer, Supplier, Function

Exception Handling

- What is exception?
- Type of Errors
- Exception class hierarchy
- Exception Handling Mechanism
- try keyword
- Catch keyword
- Checked and unchecked exceptions
- Throw vs throws keyword
- Finally block
- Multi catch
- Final re-throw
- Try-with-resources (ARM)
- User-defined exception

GUI Development using AWT & Swing

AWT

- What is AWT?
- Why AWT?
- Features of AWT
- AWT Component and its hierarchy
- Layout Managers
- Steps in AWT Program
- Event Handling

- What is an event?
- EventListener
- Adapter classes
- Event Delegation Model
- Demo Program using actionPerformed()

SWING

- What is Swing?
- AWT vs Swing
- JFC – Java Foundation Classes
- Hierarchy of Java Swing classes
- MVC Architecture
- Event Handling in Swing
- Demo Program using JFrame with event handling

Multithreading

- Multitasking
- Multiprocessing vs Multithreading
- What is thread?
- Thread Life Cycle
- Creating thread using java.lang.Thread class
- Creating thread using java.lang.Runnable interface
- Thread class and it's methods
- Inter thread communication
- Thread Synchronization
- synchronized keyword
- synchronized method & synchronized block

Collection Framework & Generics

- Collection and it's types
- java.util package
- Generics Syntax
- List – ArrayList, Vector
- Set – HashSet, SortedSet, NavigableSet – TreeSet
- Map – HashMap, SortedMap, NavigableMap – TreeMap

- Hashtable and Properties class
- Importance of equals() & hashCode() methods
- Searching & Sorting Algorithm Arrays & Collections classes
- Comparable vs Comparator interfaces

Stream API

- What is Stream?
- Types of Stream – Sequential & Parallel Stream
- java.util.stream package
- Stream operations – intermediate and terminal operations
- map(), reduce(), filter(), forEach(), limit(), skip() methods
- Collectors – toCollection(), toList(), toSet(), toMap() methods

File IO & Object Serialization

- java.io package
- File class & it's method
- File Reading, Writing and Appending Operation
- Byte Stream – FileInputStream & FileOutputStream
- Character Stream – FileReader & FileWriter
- BufferedReader & BufferedWriter, PrintWriter
- Autocloseable
- Using try-with-resources
- Non-blocking IO (nio)

- java.nio.file package
- Path interface
- Files class, Paths class
- FileSystem
- Object Serialization
- ObjectInputStream & readObject() method
- ObjectOutputStream & writeObject() method
- transient keyword

Networking

- java.net package
- URL
- Socket Programming
- TCP vs UDP
- Socket, ServerSocket
- DatagramSocket, DatagramPacket

Unit Testing

- Introduction
- What is Testing?
- Why Unit Testing?
- Testing Terminologies
- Junit Framework
- Junit Annotations
- Creating Test Cases

Adv Java

Duration: 2 months

Introduction to SQL

- | | |
|--|--|
| <ul style="list-style-type: none"> • What is SQL? • RDBMS • Installation of MySQL Database • SQL Data Types • SQL Constraints • Connecting to MySQL • Creating Database | <ul style="list-style-type: none"> • Performing CRUD operations • INSERT Query • SELECT Query with WHERE Clause • UPDATE Query • DELETE Query |
|--|--|

JDBC

- Java Database Connectivity
- java.sql package
- Connection interface
- Driver interface
- Types of JDBC Drivers
- Driver Manager Class
- Connecting to MySQL using JDBC API
- Statement
- Prepared Statement
- Callable Statement
- Result Set
- Types of Result Set – Forward Only, Scrollable
- Result Set Meta Data
- Database MetaData
- javax.sql.rowset package
- What is RowSet?
- Types of RowSet – JdbcRowSet vs CachedRowSet

XML

- What is XML?
- Extensible Markup Language
- XML vs HTML
- XML Syntax Rules
- Creating simple XML
- DTD – Document Type Definition
- XML Schema

Maven

- What is Maven?
- Features of Maven
- Environment Setup
- Build Life Cycle
- Phases
- Goals
- Plugins
- POM xml file
- Dependency Management
- Maven Dependencies
- Maven Repositories
- Maven Archetypes
- Creating Maven Project using Eclipse

Introduction to Web Technologies

- Distributed Architecture
- Two-tier Architecture
- Three tier (N-tier) Architecture
- Web Application
- Web Application Architecture
- Web Container
- Web Server
- Web Browser
- HTTP – Stateless Protocol
- HTTP Methods – GET, POST, PUT, DELETE, etc
- HTTP Message Format
- Request Message Format – Request Header, Request Body

- Response Message Format – Response Header , Response Body
- Client Side Programming
- Server Side Programming

Servlet

- Introduction to Servlet
- Servlet API – Servlet, ServletConfig interfaces
- GenericServlet
- HttpServlet
- Servlet Life Cycle
- ServletContext vs ServletConfig
- Servlet Configuration - web.xml
- @WebServlet Annotation
- Creating Servlet
- Configuring Tomcat Server in Eclipse IDE
- Deploying and Running Servlet on Tomcat via Eclipse
- Inter Servlet Communication
- Request Dispatcher – forward(), include()
- Response Redirection
- What is Session?
- Session Tracking
- Hidden Form Field
- Cookie
- HttpSession
- URL Rewriting

JSP

- Introduction to Java Server Pages
- JSP API
- JSP Life Cycle
- Creating and Running JSP Page on Tomcat Server
- Building Blocks of JSP
- JSP Directives
- Types of JSP Directives
- Page Directive and it's attributes
- Include Directive
- JSP Implicit Objects
- JSP Scripting Elements
- Declaration Script
- Scriptlet
- JSP Expression
- Java Bean Standards
- JSP Standard Actions
- Java Bean Actions – useBean, setProperty, getProperty
- forward and include actions
- include directive vs include action
- Custom Tag
- Taglib directive

JSTL

- Java Standard Tag Library
- Types of JSTL sub-libraries
- Core JSTL tags
- Formatting JSTL tags
- SQL JSTL tags
- XML JSTL tags
- JSTL Function tags
- Demo – Core Tag

Web Service

- What is Web Service?
- Web Service vs Web Application
- SOA – Service Oriented Architecture
- Service Provider
- Service Consumer
- Types of Web Service
- SOAP Web Service
- RESTful Web Service
- ROA – Resource, URI and Representation
- Uniform Interfaces
- Creating REST API
- Running RESTful Web Service on Tomcat
- Consuming REST API

Java Framework - Hibernate JPA

Duration: 1 month

Prerequisites: Core Java, Collection Framework, JDBC, XML, SQL, RDBMS

Object Relationship Mapping

- ORM Principle
- Drawbacks of JDBC
- Benefits of ORM
- Paradigm Mismatch
- Types of ORM Frameworks

Hibernate JPA Introduction

- What is Hibernate?
- Features of Hibernate
- Advantages of Hibernate
- Hibernate in Layered Application Design
- Hibernate Architecture
- Components of Hibernate
- Configuration
- Session Factory

- Session and it's methods –
get(), load(), save(), update(), delete()
- What is JPA?
- Hibernate and JPA

Simple Class Mapping

- Entity Class
- POJO class and it's advantages
- Hibernate JPA Mapping
- Mapping POJO class with Database Table
- JPA Annotations - @Entity, @Table, @Id, @Column
- Hibernate Configuration
- hibernate.cfg.xml file
- Configuring JDBC Driver, JDBC URL
- Hibernate Dialect
- Creating Hibernate Application
- Performing CRUD operations

Component Mapping

- Mapping 'HAS A' relationship
- Creating Contained class
- Creating Container class
- Using JPA Annotations -
@Embedded, @Embeddable

Inheritance Mapping

- Types of Inheritance Mapping
- Table Per Class Hierarchy
- JPA Annotations - @Inheritance
- Inheritance Strategy – InheritanceType.
- SINGLE_TABLE
- @DiscriminatorColumn,
- @DiscriminatorValue
- Table Per Concrete Class
- Inheritance Strategy – InheritanceType.
- TABLE_PER_CLASS
- Table Per Joined Subclass
- Inheritance Strategy – InheritanceType.
- JOINED
- @PrimaryKeyJoinColumn

Hibernate JPA Entity Life Cycle

- Transient State
- Persistent State
- Detached State

JPQL

- Java Persistent Query Language
- From clause
- Where clause
- Order By
- Select clause
- Using Criteria

Relationship Mapping

- One-to-One Mapping
- @OneToOne, mappedBy
- @JoinColumn
- One-to-Many Mapping
- @OneToMany
- Many-to-One Mapping
- @ManyToOne
- Many-to-Many Mapping

Java Framework – Spring & Spring Boot

Duration: 1.5 month

Prerequisites: Core Java, Collection Framework, JDBC, XML, SQL, RDBMS, Hibernate JPA, Maven, Servlet, JSP, RESTful Web Service,

Introduction to Spring

- Mapping 'HAS A' relationship
- Creating Contained class
- Creating Container class
- Using JPA Annotations -
@Embedded, @Embeddable

Spring Architecture

- Types of Spring Module
- Spring IoC Container
- Spring Core - BeanFactory
- Spring Context - ApplicationContext
- Spring AOP
- Spring DAO - JdbcTemplate
- Spring ORM - HibernateTemplate
- Spring Web
- Spring Web MVC

Spring Bean and Configuration

- Creating Spring Bean
- Spring Bean Configuration
- Xml Based Configuration
- Annotation Based Configuration
- Stereotype Annotation - @Component
- @Configuration
- @ComponentScan
- JavaConfig - @Bean
- Bean Scoping
- Bean Scopes – singleton, prototype, request, session
- @Scope Annotation

Spring Bean and Configuration

- Spring Aware Interfaces
- BeanNameAware
- BeanFactoryAware
- ApplicationContextAware
- InitializingBean – afterPropertiesSet()

- @PostConstruct
- Custom Init Method
- DisposableBean
- Custom Destroy Method

Dependency Injection

- Constructor Injection
- Constructor-arg
- Setter Injection
- Property
- Bean Autowiring
- @Autowired
- @Qualifier
- Method Injection
- Collection Injection
- List Injection
- Set Injection
- Map Injection
- Properties Injection

Spring Boot

- Introduction to Spring Boot
- Features of Spring Boot
- Spring Boot Starter Dependencies
- @SpringBootApplication
- AutoConfiguration
- Creating Spring Boot Application
- Spring Initializr – <https://start.spring.io/>

Spring Data

- What is Spring Data?
- Features of Spring Data
- Spring Data JDBC
- Repository interface and it's hierarchy
- Creating custom Data Repository interface
- Performing CRUD Operations
- Custom methods
- @Query, @Param

Spring Data JPA

- Jpa Repository interface
- Creating custom Data JPA Repository interface
- Performning CRUD Operations
- Customer methods
- JPQL Queries
- Transaction Configuration

Spring Web

- Spring Web MVC
- MVC Architecture
- Spring Web Architecture
- DispatcherServlet
- HandlerMapping
- View Resolver
- Controller - Request Handler
- @Controller Stereotype Annotation
- @RequestMapping

RESTful Web Service

- @RestController
- @GetMapping
- @PostMapping
- @PutMapping
- @DeleteMapping
- Creating REST API

Spring Security

- Authentication
- Authorization
- Roles
- Permissions
- Features of Spring Security
- Authentication Models
- Password Encoders
- User Store - Types
- OAuth2

Your Success Is Our Priority:

Every student at **SevenMentor** gets personalized guidance, Mentorship, and ample opportunities to address individual questions and concerns. All our sessions are designed to be engaging, interactive, and tailored to your learning pace, ensuring you grasp each concept with clarity.



Pooja Ghodekar

Full Stack Trainer
(SevenMentor & Training Pvt. Ltd.)

A passionate trainer having 6 plus years of experience in Java Full stack technologies with a comprehensive understanding of both front-end and back-end technologies, she provides in-depth training and mentorship to nurture the next generation of full stack developers. Her commitment of staying up to date with the latest industry trends and best practices ensures that her students receive the most relevant and up-to-date education in the field.



Pooja bhavsar

Java full stack tranier & C & C++ Trainer
(SevenMentor & Training Pvt. Ltd.)

Working as Java Fullstack Trainer having 4+ years of experience in teaching field. I have trained over 500+ professionals and students about core to java frameworks, SQL, and web development. Also participated in industrial project development process.



Dharmaraj Pawale

Sr. Technical Trainer
(SevenMentor & Training Pvt. Ltd.)

Dharmaraj has expertise in retail as well as corporatetraining in Java Full Stack, MEAN Stack, MERN Stack, Hadoop-Big Data, Spark. Hehas extensive 19+ years experience in both Software Training and Development.He has trained more than 10000+ students and corporate professionals in India aswell as overseas. He is also Salesforce Certified Administrator. His area ofexpertise lies in both Frontend as well as Backend Technologies. Frontend technologiesinclude HTML5, CSS3, Bootstrap, JavaScript, React & Redux, Angular andBackend Technologies include Java, Servlet & JSP, RESTful Web Service, HibernateJPA, Spring, Spring Boot, MicroService, Node JS, Express JS, RDBMS Databasesuch as MySQL, Oracle SQL, PL/SQL, NoSQL Databases such as MongoDB



Onkar Nagarkar

Java Full stack tranier

(SevenMentor & Training Pvt. Ltd.)

An Expert Java Trainer with over 5+ years of Corporate experience in designing and developing end-to-end industry level scalable applications. Excels in conveying core and complex programming concepts in depth in the simplest way possible, enabling any student, even from a non-IT background to crack any MNC comfortably. Knows how to engage and bring out the required confidence in a student, to master the interview rounds and communicate effectively. Skilled in Complete end-to-end software development which includes C, C++, Java, SQL, Spring boot, Angular and JavaScript. as MySQL, Oracle SQL, PL/SQL, NoSQL Databases such as MongoDB



Anil Giri

Java full stack & .Net Full stack Trainer

(SevenMentor & Training Pvt. Ltd.)

Professional IT trainer with over decade and eight years of experience in Java and .NET Full stack technologies. Worked for many corporates and delivered corporate training PAN India also managed team of 50 trainers and has many accolades under his belt due to his excellence at work. An expert at designing training courses with technical content, Effective at imparting knowledge through his superior verbal communication skills, active listening style and unique teaching pedagogy. He has proven success in utilizing cutting-edge technology to devise training exercises for all types of learners.

THINGS THAT SET US APART FROM THE REST:

SevenMentor prepares you well so that you can embark on a rewarding career in Java Programming through our training.



Hands-On Projects: Gain practical experience by working on real-world projects, building a robust portfolio that will impress potential employers.



Flexibility: Our flexible schedule options allow you to learn at your own pace, making it perfect for both beginners and experienced developers looking to upskill.

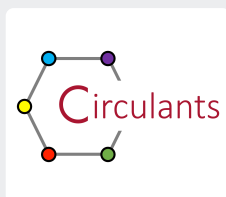
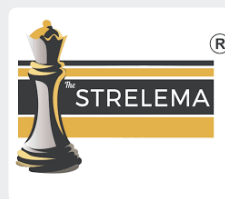


Career Support: We're dedicated to your success! Benefit from career guidance, resume building, interview prep, and job placement.



Community: Join a vibrant community of like-minded learners, where you can collaborate, share ideas, and network with peers.

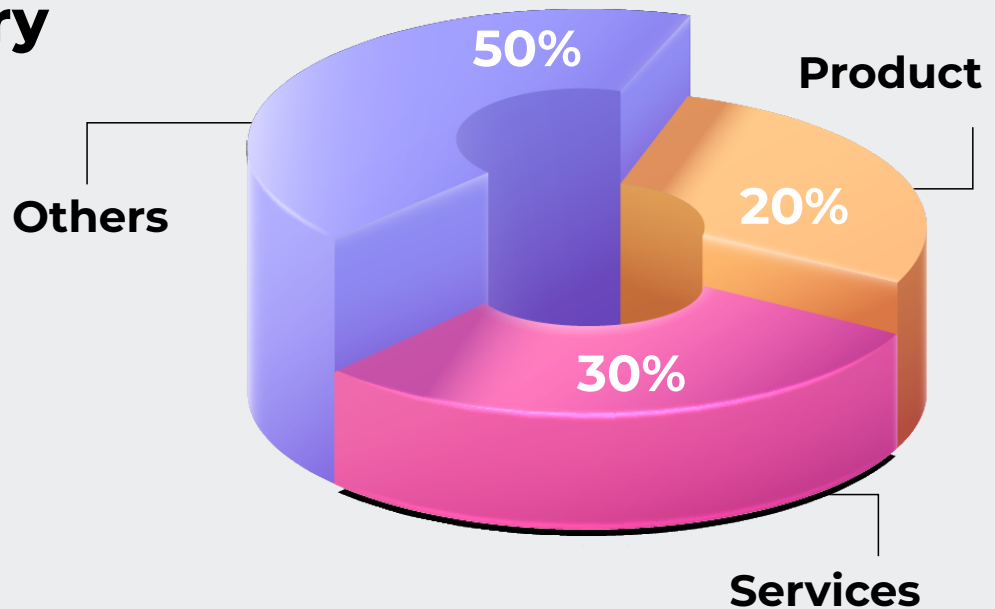
Our Students are at reputed Tech Companies



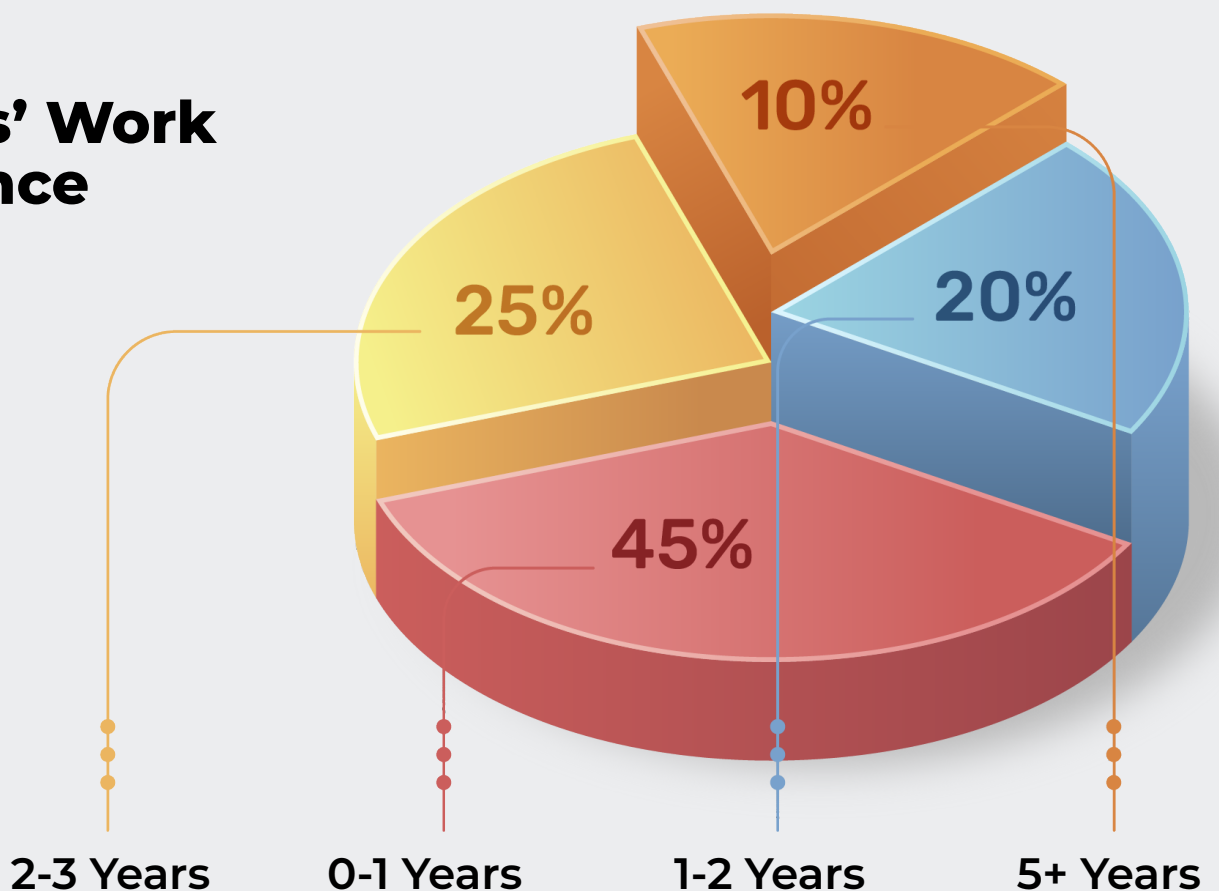
Success is around the corner!

Consider the potential earnings and opportunities that a career in Java Programming can offer. With Java Programming you'll be well-positioned to secure a job that provides you with a stable and potentially prosperous future.

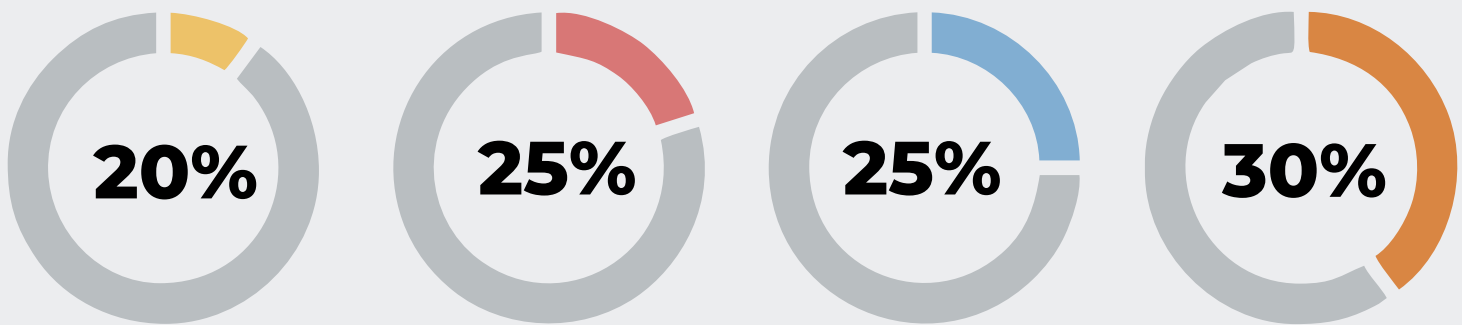
Learners' Industry Background



Learners' Work Experience



Learners' Expertise



Begin Your Journey Towards Better Future:

The global Java Programming vacancies will grow by 56% and generate 4 million new jobs by 2030.

So get ready to dive headfirst into the dynamic world of Java Programming by gaining all round experience at SevenMentor

Affordable Training without Compromise:

We understand that pursuing your dreams in Java Programming shouldn't be a financial burden. That's why we've made affordability a cornerstone of our Java Programming Course. We believe that quality education should be accessible to everyone, and we've structured our program to reflect this commitment. Our Java Programming Course takes you on an exciting journey but at a substantially low price.

How to become a great JAVA Developer?

- Enroll at **SevenMentor Institute**
- Get hands-on training from **experienced teachers**
- Receive Industry-recognized **JAVA Programming Certification**
- Work for leading **MNCs** through our **on-campus interviews**

Success is Just a Call Away!

So if you are ready to code your way to success?
Enroll now at SevenMentor and unlock your potential
as a Java Programming. Our counsellors are a call
away and they will be more than happy to talk with
you. Anyday, Everyday, we are there for you!

Request For Call Back



020 7117 3125