Java Professional

Course Code: LFH/Java/04

Duration: 100 hours

Course Syllabus

INTRODUCTION

This industry oriented course is developed by both the Software development division & Training division of **ipsr solutions limited**. IPSR is **a public limited IT company** with 20 years of expertise in Software product development, Training services, Placement services & Digital Marketing services. During the past 2 decades, IPSR has trained candidates from **50+ countries** and helped **40000+ candidates** to build their IT career. Our IT services division is a pioneer in development of **Academic solution products**, incorporating cutting edge technologies like Artificial Intelligence, Data Analytics and Machine learning. Live industry experts from this IT division contribute a major role in delivering this course. Our placement division is having **1500+ placement tie-up companies** and we are conducting recruitment on all days.

The Course curriculum is designed and developed by a team of expertise panel lead by following academicians

- ❏ **Dr. Mendus Jacob, M.Sc., M.Phil., Ph.D., MIoD**
  - ❏ M.D & C.E.O - IPSR & Valin Technologies, U.K.
  - ❏ Director - MCA, Marian College, Kuttikkanam (Autonomous)
  - ❏ Former Director of School of Applicable Mathematics, M.G. University.
  - ❏ Academician and Entrepreneur with 30+ years experience
- ❏ **Dr. Sunil Job K.A, M.Sc, M.Ed, M.Phil, Ph.D., RHCE**
  - ❏ Chief of Academic Solutions - IPSR
  - ❏ Former college Principal and a Specialist in Data Analytics & Machine Learning
  - ❏ Blogger and a Resource person for National conferences
  - ❏ Academician with 25+ years experience

What you'll learn

Java Fundamentals, OOPS, Arrays, Strings, Generics & Collections, Exception Handling & Assertions, Functional Interface and Lambda Expressions , Migration to a Modular Application, Java Stream API, I/O Fundamentals and NIO2, Concurrency, Database programming with JDBC, Web Development using HTML, CSS, JavaScript, JSP, Servlets and Hibernate, MVC development using Spring Framework

Description

The "Java Professional" course covers the skills required by a Java Programmer, Web developer, MVC Developer etc.

Course Outcome (CO)

While successfully completing this course, the learner will be able to:

➔ Create Java standalone applications.
➔ Create simple web applications using Java web components such as JSP, Servlets and ORM as Hibernate.
➔ Create an MVC application using Spring Framework

What does this course give you?

The "Java Professional" course covers the skills required by a Java Programmer,

Web developer, MVC Developer etc.

Course content

Java SE Programmer

1. **Understand Java Technology and Environment**

   ● Describe Java Technology and the Java development environment

   ● Identify key features of the Java language

2. **Create a simple java program**

   ● Create an executable Java program with a main class

   ● Compile and run a Java program from the command line

   ● Create and import packages

3. **Working with Java Primitive Data Types**

   ● Declare and initialize variables (including casting and promoting primitive data types)

   ● Identify the scope of variable

   ● Use local variable type inference

4. **Using Operators and Decision Constructs**

   ● Use Java operators including the use of parenthesis to override operator precedence

   ● Use Java control statements including if, else, and switch

   ● Create and use do/while, while and for loops, including nested loops, use break and continue statements

5. **Describing and Using Objects and Classes**

   ● Declare and instantiate Java objects, and explain objects' lifecycles (including creation, dereferencing by reassignment, and garbage collection)

   ● Define the structure of a Java class

   ● Read or write to object fields

6. **Creating and Using Methods**

   ● Create methods and constructors with arguments and return values

   ● Create and invoke overloaded methods

   ● Apply the static keyword to methods and fields

7. **Applying Encapsulation**

   ● Apply access modifiers

   ● Apply encapsulation principles to a class

8. **Reusing Implementations Through Inheritance**

   ● Create and use subclasses and superclasses

   ● Create and extend abstract classes

   ● Enable polymorphism by overriding methods

   ● Utilize polymorphism to cast and call methods, differentiating object type versus reference type

   ● Distinguish overloading, overriding, and hiding

9. **Working with String APIs**

   ● Create and manipulate Strings

   ● Manipulate data using the StringBuilder class and its methods

10. **Working with Java Arrays**

    ● Declare, instantiate, initialize and use a one-dimensional array

- Declare, instantiate, initialize and use two-dimensional array
- Use for each loop

11. **Final & Nested Classes**
    - Create and use final classes
    - Create and use inner, nested and anonymous classes
    - Create and use enumerations

12. **Programming Abstractly Through Interfaces**
    - Create and implement interfaces
    - Distinguish class inheritance from interface inheritance including abstract classes
    - Create and use interfaces with default methods
    - Create and use interfaces with private methods

13. **Exception Handling and Assertions**
    - Describe the advantages of Exception handling and differentiate among checked, unchecked exceptions, and Errors
    - Create try-catch blocks and determine how exceptions alter program flow
    - Create and invoke a method that throws an exception
    - Use the try-with-resources construct
    - Create and use custom exception classes
    - Test invariants by using assertions

14. **Generics and Collections**
    - Use wrapper classes, autoboxing and autounboxing
    - Create and use generic classes, methods with diamond notation and wildcards
    - Describe the Collections Framework and use key collection interfaces
    - Use Comparator and Comparable interfaces
    - Create and use convenience methods for collections

15. **Functional Interface and Lambda Expressions**

   ● Define and write functional interfaces

   ● Create and use lambda expressions including statement lambdas, local-variable for lambda parameters

16. **Built-in Functional Interfaces**

   ● Use interfaces from the java.util.function package

   ● Use core functional interfaces including Predicate, Consumer, Function and Supplier

   ● Use primitive and binary variations of base interfaces of java.util.function package

17. **Understanding Modules**

   ● Describe the Modular JDK

   ● Declare modules and enable access between modules

   ● Describe how a modular project is compiled and run

18. **Migration to a Modular Application**

   ● Migrate the application developed using a Java version prior to SE 9 to SE 11 including top-down and bottom-up migration, splitting a Java SE 8 application into modules for migration

   ● Use jdeps to determine dependencies and identify ways to address the cyclic dependencies

19. **Java Stream API**

   ● Describe the Stream interface and pipelines

   ● Use lambda expressions and method references

20. **Lambda Operations on Streams**

   ● Extract stream data using map, peek and flatMap methods

- Search stream data using search findFirst, findAny, anyMatch, allMatch and noneMatch methods
- Use the Optional class
- Perform calculations using count, max, min, average and sum stream operations
- Sort a collection using lambda expressions
- Use Collectors with streams, including the groupingBy and partitioningBy operations

21. **I/O (Fundamentals and NIO2)**
- Read data from and write console and file data using I/O Streams
- Use I/O Streams to read and write files
- Read and write objects by using serialization
- Use the Path interface to operate on file and directory paths
- Use the Files class to check, delete, copy or move a file or directory
- Use the Stream API with Files

22. **Concurrency**
- Create worker threads using Runnable, Callable and use an ExecutorService to concurrently execute tasks
- Use java.util.concurrent collections and classes including CyclicBarrier and CopyOnWriteArrayList
- Write thread-safe code
- Identify threading problems such as deadlocks and livelocks

23. **DBMS – MySQL**
- MySql datatypes
- CRUD operations
- Subquery

- Join

- Aggregate functions

- Procedures

24. **Database Applications with JDBC**

- Connect to databases using JDBC URLs and DriverManager

- Use PreparedStatement to perform CRUD operations

- Use PreparedStatement and CallableStatement APIs to perform database operations

25. **Annotations**

- Describe the purpose of annotations and typical usage patterns

- Apply annotations to classes and methods

- Describe commonly used annotations in the JDK

- Declare custom annotations

26. **Localization**

- Use the Locale class

- Use resource bundles

- Format messages, dates, and numbers with Java

Java Web Development

27. **Web Development Basics**

- HTML, CSS, JavaScript

28. **Java Web Applications**

- Web servers and application servers

- Web Containers

- Web Application Structures - .war files, building tool - Maven

- Different Technologies using in Web Application

29. Web Development using JSP & Servlets

Servlet life cycle

Handling web requests in servlets

Servlet Context, Application Context, Servlet Config

InterServlet communication

Servlet filters

JSP Page Structure & Implicit objects

JSP Life cycle

Sessions & Session handling

Database communication – using JDBC and ORM using Hibernate

JSTL

jQuery & AJAX

Upload files & Images

Java MVC Web Development

30. MVC Web Development using Spring Framework
- Spring Framework Introduction and architecture
- Spring Containers
- Dependency Injection – Setter Based, Constructor Based
- Autowiring

- Spring MVC
    a. MVC Architecture
    b. Create Models, Views, Controllers
    c. MVC Form Tags
    d. MVC CRUD Operations
    e. MVC JPA
    f. File/Image Uploading
    g. Spring RESTFul application
    h. Spring + React JS

Project: Java MVC Web Project using Spring

      a. MVC CRUD Operations

      b. MVC JPA + Hibernate

      c. File/Image Uploading

31. Spring RESTFul application

32. Spring micro and monolithic designs

33. Introduction to Thymeleaf – Front End Development

34. Spring + Angular

Project: Java MVC Web Project using Spring

Contact Us

IPSR SOLUTIONS LTD.

Merchant's Association Building

M.L. Road, Kottayam - 686001

Kerala, India, Pin-686001

Phone: +91-481 2561410, 2561420, 2301085

Mobile: +91 9447294635, +91 9447169776

Email: training@ipsrsolutions.com

Website: http://www.ipsr.org

Learn_from_Home Portal: https://lms.ipsr.org

We have branches at Kochi, Thiruvananthapuram, Calicut and Bengaluru.