

Basic to Advance Java Course

❖ **Programming Basics**

- ❖ Fundamentals of Computer
- ❖ Understanding Applications
- ❖ Using Windows Explorer & File Structure
- ❖ Number Systems
- ❖ Application Software

❖ **Operating System & Networking**

- ❖ Need for operating system
- ❖ Functions of Operating System
- ❖ Process Management
- ❖ Memory Management
- ❖ File Management
- ❖ Device Management
- ❖ Types of operating system
- ❖ TCP/IP fundamentals
- ❖ Networks and computer networks
- ❖ 2 tier, 3 tier and n-tier architecture

❖ **Core Java**

- ❖ Introduction to Java
- ❖ History of Java
- ❖ Features of Java

- ❖ **Overview of Java**

- ❖ OOP's Concept
- ❖ Data types and Variables
- ❖ Control Structures
- ❖ Strings, Arrays

- ❖ **Objects and Classes**

- ❖ Object, Classes and Methods
- ❖ Method Overloading
- ❖ Constructors
- ❖ Object class

- ❖ **Inheritance**

- ❖ Types of Inheritance
- ❖ Method Overriding
- ❖ Dynamic method dispatch

- ❖ **Packages and Interfaces**

- ❖ Defining Packages

- ❖ **Exception Handling**

- ❖ Fundamentals of Exception Handling
- ❖ Exception types

- ❖ Try and Catch and finally
- ❖ Throw, throws
- ❖ Custom Exception

- ❖ **Inner Class and Wrapper classes**
- ❖ Inner Classes
- ❖ Static Nested Classes
- ❖ Wrapper Classes
- ❖ Anonymous Inner Classes

- ❖ **String Handling**
- ❖ Creating Strings
- ❖ String handling methods
- ❖ String Buffer and String Builder

- ❖ **Input and Output in Java**
- ❖ Byte streams & Character streams
- ❖ File
- ❖ Serialization

- ❖ **Collections Framework**
- ❖ Collection Interfaces and Classes
- ❖ Iterators

- ❖ Comparators

- ❖ **Multithreading**

- ❖ Basics of java thread
- ❖ The Thread Scheduler
- ❖ Naming a thread, Daemon thread
- ❖ Perform single /multiple tasks by multiple threads
- ❖ Thread Lifecycle and methods
- ❖ Callable Interface and its uses

- ❖ **Reflection API**

- ❖ Overview of Reflection
- ❖ Use of newInstance() method and determining the class Object
- ❖ Accessing a private method or member from outside the class

- ❖ **Java 8 and above features**

- ❖ Lambda expressions and functional interface
- ❖ Enhancements along with latest versions

- ❖ **Basic SQL**

- ❖ Introduction to Database
- ❖ Tables and SQL
- ❖ DDL and DML

- ❖ Clauses
- ❖ DB inbuilt Functions

- ❖ **Advance Java**
- ❖ **JDBC**
- ❖ Common JDBC components
- ❖ Steps to connect to the database using MySQL and Oracle database
- ❖ Types of JDBC statements – statements, prepared Statement, and callable statement
- ❖ Transactions management in JDBC
- ❖ CRUD operation using prepared Statement
- ❖ Calling Stored procedure using callable statement
- ❖ How Result Set Works in JDBC

- ❖ **Introduction to Frontend**
- ❖ HTML and CSS
- ❖ What is Javascript?
- ❖ Use of Javascript in Webpage
- ❖ Webpage Lifecysle

- ❖ **Servlets**
- ❖ Servlet – environment setup
- ❖ ServletConfig and ServletContext parameters
- ❖ Servlet life cycle

- ❖ Servlet Attribute and Servlet Parameters
- ❖ Exploring Deployment Descriptor (web.xml)
- ❖ Session Tracking In Servlet
- ❖ Servlet Filter
- ❖ Types Of Filter
- ❖ Servlet Filter Mapping In Web.Xml
- ❖ Servlet-listener
- ❖ Various Servlet Listener
- ❖ Servlet Listener Configuration
- ❖ servlet-security
- ❖ Details about MVC architecture
- ❖ Practical exposure on MVC using JSP and Servlet

- ❖ **JSP 2.X**
- ❖ JSP – Overview
- ❖ JSP – Life Cycle
- ❖ SP-Scripting Elements
- ❖ JSP- Directive Elements
- ❖ Types Of JSP Scopes
- ❖ JSP – Custom Tags
- ❖ JSP-Implicit Objects
- ❖ JSP – Exception Handling

- ❖ **Spring**
- ❖ Introduction To Spring Framework
- ❖ Spring-Environment Setup
- ❖ Steps To Use Spring Framework In Applications
- ❖ Understanding IOC And Dependency Injection
- ❖ Working with Bean Factory and Application Context
- ❖ Bean Definition
- ❖ Working with multiple configuration files
- ❖ Most popular namespaces e.g. beans,
- ❖ Best practices when working with namespaces
- ❖ Advanced XML Dependency Injection
- ❖ Externalizing constant values into properties files
- ❖ Bean Scopes and Bean Life Cycle
- ❖ Bean Post Processors
- ❖ Bean Definition Inheritance
- ❖ Dependency Injection
- ❖ Injecting Inner Beans
- ❖ Injecting Collection Type
- ❖ Annotation-Based Dependency Injection
- ❖ Autowiring and component scanning
- ❖ Annotation Based Configuration(@Required,
- ❖ @Autowired,@Qualifier)
- ❖ JSR-250 Annotations(@Resource,@PostConstruct
- ❖ ,@PreDestroy)

- ❖ Component and Stereotype Annotations
- ❖ Using MessageSource To Get Text From Property Files

- ❖ **Data Access and JDBC with Spring**
- ❖ Introduction to Spring JDBC
- ❖ How Spring integrates with existing data access technologies
- ❖ Spring JDBC APIs
- ❖ Spring JDBC development
- ❖ Data Access Exception hierarchy
- ❖ Result transformations
- ❖ Implementing Row Mapper
- ❖ Parameter mapping
- ❖ Named Parameter JDBC Template
- ❖ Transactions management in Spring
- ❖ Spring Declarative Transactions Management
- ❖ Spring Programmatic transaction management

- ❖ **Working with Spring MVC**
- ❖ Introduction to Spring MVC framework
- ❖ Creating many Spring MVC Web Applications
- ❖ Writing an annotation-based controller class- @Controller,@RequestMapping
- ❖ @PathVariable annotation
- ❖ Handling an HTML form using @RequestParam annotation

- ❖ Understanding @ModelAttribute Annotation
- ❖ Data Binding with Date, Collection
- ❖ Data Binding with a User-Defined Type, BindingResult
- ❖ @InitBinder annotation, WebDataBinder, CustomDateEditor
- ❖ Form Validations
- ❖ Form Validation (customizing error messages using Spring MessageSource)

- ❖ **Hibernate**
- ❖ Overview of Hibernate
- ❖ Hibernate-environment setup
- ❖ Object-relational mapping (ORM)
- ❖ Features of Hibernate
- ❖ Using JPA and Hibernate annotations
- ❖ Hibernate object states(Transient, Persistent and Detached Objects)
- ❖ CRUD operation in Hibernate
- ❖ Annotation
- ❖ Hibernate Mapping XML Configuration
- ❖ Hibernate Caching
- ❖ Transactions management in Hibernate
- ❖ BLOB Object

- ❖ **Aspect-Oriented Programming (AOP) with Spring**

- ❖ Introduction on Spring AOP
- ❖ Aspect-oriented programming concepts
- ❖ Integration with Spring IoC
- ❖ Defining pointcut expressions
- ❖ Implementing an advice: @Around, @Before, @After, and so on
- ❖ AspectJ APIs and annotations

- ❖ **Demonstration On Live Project.**

- ❖ **Resume Preparation & Mock Interview.**

Total Duration 35 hours

Weekday Timings (Mon-Fri) : 1 hours (1.5 Month)

Weekend Timings (Sat-Sun) : 2 hours (2 Month)

--- THANK YOU ---