

# COURSE OUTLINE

---

## JAVA Programming (JVP)

### Overview

This Java Programming course focuses on the fundamentals of the Java language, object oriented development and the structure of Java applications. It covers the Java Platform up to version 8.

This hands-on course shows delegates how to develop Java systems that use various core API packages. They will develop programs that support multi-threading and networking. Java has always been associated with GUI development and this course allows delegates to develop graphical applications using Swing and JacaFX. They also learn how to access databases and deploy applications..

### Duration

5 Days

### Target Audience

This Java Programming course is aimed at anyone who needs to learn how to develop in Java

### Course Outcomes

By the end of the course delegates should be able to:

- Explain the Java architecture
- List and describe the standard Java packages in various versions
- Choose between various design strategies
- Understand the benefits of design patterns
- Create, build, and debug Java projects
- Build and release Java applications
- Store objects using Java's Collection framework

# COURSE OUTLINE

---

- Write robust applications using Exception handling
- Perform a variety of I/O using Stream and File classes
- Format strings, dates and numbers
- Use Java localisation features
- Develop GUI systems using the Swing classes
- Raise and respond to events
- Create Java programs with concurrent threads
- Create Network aware applications
- Access databases using JDBC

## Course Contents

### Course Introduction

- Administration and Course Materials
- Course Structure and Agenda
- Delegate and Trainer Introductions

### INTRODUCING JAVA

- The Java Development Environment
- The Java Runtime Environment
- Licensing and OpenJDK
- Compiling and Executing Java Programs
- Java Program types
- Java SE and Java EE
- Integrated Development Environments

### THE JAVA LANGUAGE

- Variables

# COURSE OUTLINE

---

- Data Types
- Primitive and Reference Variables
- Declaring, Initializing and Releasing Variables
- Arithmetic and Comparison Operators
- Shortcut Operators
- Converting Data Types
- Statements and Semi-Colons
- IF ... ELSE and SWITCH CASE statements
- Loops
- Comments
- Arrays

## **OBJECTS AND CLASSES**

- Objects
- Encapsulation
- Classes
- Methods
- Strings
- Packages
- Class Member Data
- Defining and Using Constructors
- Static Initializers
- Static Imports
- Garbage Collection
- Object Lifetime

# COURSE OUTLINE

---

- Wrapper Classes
- Enumerations
- Inner Classes, Named, Static and Anonymous

## **INHERITANCE**

- Classes and Inheritance
- Methods in the Sub Class
- Constructors
- Polymorphism
- Final Classes
- Protected Modifier
- Super keyword and Object class
- Converting Reference Types

## **ABSTRACT CLASSES AND INTERFACES**

- Abstraction
- Abstract Classes
- Abstract Class References
- Abstract Methods
- Abstraction Rules and Guidelines
- Interfaces
- Defining and Implementing an Interface
- Type Conversions and Interfaces
- Lambda Expressions

## **DESIGN CONSIDERATIONS**

- Composition

# COURSE OUTLINE

---

- Method Delegation
- Introduction to Design Patterns
- Pattern Examples - Delegate, Singleton, DAO and Decorator

## GENERIC AND ANNOTATIONS

- User defined Annotations
- Built in Annotations
- Generics
- The Enhanced FOR Loop
- Variable Method Arguments
- Auto Boxing

## COLLECTIONS

- The java.util and java.util.stream Packages
- Iterator
- Lists
- Maps
- Sets
- Stack and Deque
- Diamond Syntax (Type Inference)
- How to Choose a Collection Class

## EXCEPTIONS AND ASSERTIONS

- Syntax Errors
- Runtime Errors
- Logical Errors
- Exception Handling

# COURSE OUTLINE

---

- Exception Objects
- Handling Exceptions
- Try-with-resources
- Multi-catch
- User-Defined Exception Classes
- Exception Handling Guidelines
- Assertions

## **FORMATTING STRINGS, DATES AND NUMBERS**

- String Immutability
- String Class Methods
- The StringBuffer and StringBuilder Classes
- Text Input and Parsing
- Text Formatting
- Using Regular Expressions
- Date Formatting
- Calendar and java.time package
- Number Formatting
- Localization
- Resource Bundles

## **SWING AND JAVAFX**

- AWT: The Precursor to Swing
- The Java Event Model
- Swing Overview
- JAVAFX Overview

# COURSE OUTLINE

---

- Create a JAVAFX GUI
- Event handling
- Using Scene Builder
- Visual effects
- Model View Controller (MVC) Design Pattern

## FILE AND NETWORK INPUT/OUTPUT

- Byte Oriented Input and Output Streams
- Decoration
- Character Oriented Streams
- Standard I/O Streams
- Examples Using I/O Streams
- Java and URLs
- Handling Files and Directories
- The Path Interface
- Files and Paths class and java.nio.file package
- Reading and Editing Files
- Monitoring with WatchService

## OBJECT SERIALIZATION

- Introduction
- Examples
- Serialization Issues
- The Externalizable Interface
- Version Serialized Files

# COURSE OUTLINE

---

## THREADS AND CONCURRENCY

- What are Threads?
- Creating Threads
- Implementing the Runnable Interface
- Synchronizing Access to Resources
- Communicating Between Threads
- Immutable Objects
- Using the `java.util.concurrent` Package
- Executors and ThreadPools

## DATABASE CONNECTION WITH JDBC

- The JDBC API
- Adding JDBC Libraries
- Registering the JDBC Driver
- Connecting to the Database
- Creating and Executing Statements
- The `PreparedStatement` and `CallableStatement` Classes
- Using `RowSet` Classes
- Implementing the Data Access Object Pattern

## DEPLOYMENT WITH JAR, EAR AND WAR FILES

- Creating Jar Files
- The Jar Manifest
- WAR Files
- EAR Files



# COURSE OUTLINE

---