

Java Programming

5 Days

COURSE OVERVIEW

This 5-day course is a comprehensive introduction to the JAVA programming language. The course covers all the main areas of JAVA from the concepts of Object Oriented Programming to the creation of classes, members and methods.

Other areas that are also addressed include Class Loading, Inheritance, Polymorphism, Abstract Classes, Interfaces, Arrays, Streams, Collections, Exceptions, Multi-threading, JDBC and GUI design.

Throughout this course, extensive hands-on exercises, performed under the expert tuition of an experienced instructor, provide delegates with practical experience.

COURSE OBJECTIVES

By the end of the course attendees will:

- Know the concepts of object-oriented programming
- Know how to set up a development environment
- Know the differences between objects and primitive types.
- Be familiar with the terms abstraction, encapsulation, inheritance and polymorphism
- Create classes that implement inheritance and polymorphism
- Understand identifiers, keywords, types and flow control
- Know how exceptions work in JAVA
- Create text-based applications
- Develop GUIs using Swing
- Understand multithreading, I/O streams, channels and networking
- Know about the different types of Collections
- Write an application that interacts with a database

COURSE CONTENT

The Java Platform

- What is Java?
- The Object Oriented Approach
- The Java Virtual Machine (JVM)
- Java Security
- JDK Components

Object Oriented Principles

- Classes, Objects and Methods
- Instantiation
- Abstraction
- Inheritance
- Encapsulation

- Aggregation and Polymorphism
- UML Techniques and Notation
- The Java Class Hierarchy

Basic Java Syntax and Coding Conventions

- How to Set Up the JAVA Environment
- The Mandatory “Hello World” Example
- Compilation and Runtime
- Defining Classes
- Specifying Methods
- Using Your Classes
- Java File Structure
- Using JavaDoc

Primitive Data Types and Operators

- Declaring and Initialising Variables
- Data Types
- Operator Categories
- Implicit and Explicit Casting
- String Object Literals and the Concatenation Operator

Controlling Program Flow

- Decisions
- Loops
- Break and Continue

Creating Classes and Objects

- Instance and Class Variables
- Instance and Class Methods
- Creating Instance Variables and Methods
- Constructor Methods
- Creating Class Variables and Methods

Class Loading, Object Life Cycle and Inner Classes

- Method Overloading
- Constructors and the this keyword
- Access Modifiers
- Defining Inner Classes
- Life Cycle of an Object
- Garbage Collection

Using Strings, StringBuffer, Wrapper and Formatting Classes

- The Class String
- Using the StringBuffer Class to Hold Modifiable Text
- Conversions To and From Primitive Types Using the Wrapper Classes
- The DateFormat class
- The DecimalFormat class
- The MessageFormat class

Inheritance and Polymorphism

- Inheritance Hierarchies
- Constructors and Inheritance
- Calling and Overriding Superclass Methods
- Polymorphism
- Class Casting
- Protected Variables and Methods
- Final Methods and Classes

Abstract Classes and Interfaces

- Abstract Classes and Methods
- Defining and Implementing Interfaces
- Polymorphism with Abstract Classes and Interfaces

Arrays

- Arrays of Primitive Types
- Arrays of Object References
- Processing Command Line Arguments
- The toCharArray() Method
- Multidimensional Arrays

Streams

- What Are Streams?
- The Classes for Streams
- Reading Data from the Keyboard
- File Objects
- Writing to Files
- Reading from Files
- Object Serialisation

Collections

- Sets
- Lists or Sequences
- Maps
- The Iterator Interface
- The ListIterator Interface
- Vectors
- Hashing

Exceptions

- Exceptions and Errors
- The Throwable Object
- Catching Exceptions in a try-catch Block
- Catching More Than One Exception
- The finally Block
- Rethrowing Exceptions
- Ignoring Exceptions
- Throwing Your Own Exceptions

Threads

- The Three Parts of a Thread
- How to Create Threads
- How to Initiate a Thread
- How to Stop a Thread
- Controlling Threads that Share Data

User Interface Design

- Swing Components and Containers
- Building a GUI Framework
- Layout Managers
- Examples of Swing Components

Event Handling

- Model-View-Controller Architecture
- UIManager Basics
- The Java Event Model
- Event Handlers

Accessing the Database with JDBC

- Setting PATH and CLASSPATH Variables to use JDBC Drivers
- Loading and Registering a JDBC Driver
- Connecting to an Oracle database
- Using JDBC to Execute a SELECT Statement
- Mapping Simple Database Types to Java Types
- Executing Simple DML and DDL Statements
- Controlling Transactions
- Controlling Statement Caching