# Core Java 8 +JDBC Program Duration: 15 days

**Contents:**

# Declarations and Access Control

* Identifiers
* Legal Identifiers
* Sun's Java Code Conventions
* Declare Classes
* Source File Declaration Rules
* Class Declarations and Modifiers
* Concrete Subclass
* Declaring an Interface
* Declaring Interface Constants
* Declare Class Members
* Access Modifiers
* Nonaccess Member Modifiers
* Constructor Declarations

# Object Orientation

* Encapsulation
* Inheritance, Is-A, Has-A
* Polymorphism
* Overridden Methods
* Overloaded Methods
* Reference Variable Casting
* Implementing an Interface
* Return Type Declarations
* Returning a Value
* Constructors and Instantiation
* Default Constructor
* Overloaded Constructors
* Statics
* Static Variables and Methods
* Coupling and Cohesion

# Array, Type Casting, Wrapper classes, AutoBoxing, Garbage Collection, Enums

* Stack and Heap—Quick Review
* Literals, Assignments, and Variables
* Literal Values for All Primitive Types
* Assignment Operators
* Casting Primitives
* Passing Variables into Methods
* Passing Object Reference Variables
* Does Java Use Pass-By-Value Semantics?
* Passing Primitive Variables
* Array Declaration, Construction, and Initialization
* Declaring an Array
* Constructing an Array
* Initializing an Array
* Initialization Blocks
* Using Wrapper Classes and Boxing
* An Overview of the Wrapper Classes
* Creating Wrapper Objects
* Using Wrapper Conversion Utilities
* Autoboxing
* Overloading
* Garbage Collection
* Overview of Memory Management and Garbage Collection
* Overview of Java's Garbage Collector
* Writing Code That Explicitly Makes Objects Eligible for Garbage Collection
* Enums

# Operators

* Java Operators
* Assignment Operators
* Relational Operators
* instanceof Comparison
* Arithmetic Operators
* Conditional Operator
* Logical Operators

# Flow Control, Exceptions

* if and switch Statements
* if-else Branching
* switch Statements
* Loops and Iterators
* Using while Loops
* Using do Loops
* Using for Loops
* Using break and continue
* Handling Exceptions
* Catching an Exception Using try and catch
* Using finally
* Propagating Uncaught Exceptions
* Defining Exceptions
* Exception Hierarchy
* Handling an Entire Class Hierarchy of Exceptions
* Exception Matching
* Exception Declaration and the Public Interface
* Rethrowing the Same Exception
* Common Exceptions and Errors

# Maven Fundamentals

* Introduction
* Folder Structure
* The pom.xml
* Dependencies
* Goals
* Scopes
* The Compiler Plugin
* Source Plugin
* Jar Plugin

# TDD with Junit 5

* Types of Tests
* Why Unit Tests Are Important
* What's JUnit?
* JUnit 5 Architecture
* IDEs and Build Tool Support
* Setting up JUnit with Maven
* Lifecycle Methods
* Test Hierarchies
* Assertions
* Disabling Tests
* Assumptions
* Test Interfaces and Default Methods
* Repeating Tests
* Dynamic Tests
* Parameterized Tests
* Argument Sources
* Argument Conversion
* What Is TDD?
* History of TDD
* Why Practice TDD?
* Types of Testing
* Testing Frameworks and Tools
* Testing Concepts
* Insights from Testing
* Mocking Concepts
* Mockito Overview
* Mockito Demo
* Creating Mock Instances
* Stubbing Method Calls

# Strings, I/O, Formatting, and Parsing

* String, StringBuilder, and StringBuffer
* The String Class
* Important Facts About Strings and Memory
* Important Methods in the String Class
* The StringBuffer and StringBuilder Classes
* Important Methods in the StringBuffer and StringBuilder Classes
* File Navigation and I/O
* Types of Streams
* The Byte-stream I/O hierarchy
* Character Stream Hierarchy
* RandomAccessFile class
* The java.io.Console Class
* Serialization
* Dates, Numbers, and Currency
* Working with Dates, Numbers, and Currencies
* Parsing, Tokenizing, and Formatting
* Locating Data via PatternMatching
* Tokenizing

# Generics and Collections

* Overriding hashCode() and equals()
* Overriding equals()
* Overriding hashCode()
* Collections

o So What Do You Do with aCollection?

* List Interface
* Set Interface
* Map Interface
* Queue Interface
* Using the Collections Framework
* ArrayList Basics
* Autoboxing with Collections
* Sorting Collections and Arrays (Collections.sort method)
* Navigating (Searching) TreeSets and TreeMaps

# Threads

* Defining, Instantiating, and Starting Threads
* Defining a Thread
* Instantiating a Thread
* Starting a Thread
* Thread States and Transitions
* Thread States
* Preventing Thread Execution
* Sleeping
* Thread Priorities and yield( )
* Synchronizing Code
* Synchronization and Locks
* Thread Deadlock
* Thread Interaction
* Using notifyAll( ) When Many Threads May Be Waiting

# Concurrent Patterns in Java

* Introducing Executor Service based on Executor pattern
* What Is Wrong with the Runnable Pattern?
* Callable Interface
* Defining the Executor Service Pattern, a First Simple Example
* What is Future?
* How to use Future?
* Explain need for Completable Future.

# Concurrent Collections

* Implementing Concurrency at the API Level
* Hierarchy of Collection and Map, ConcurrentInterfaces
* Why should you use Concurrent Collections?
* ConcurrentHashMap
* Understanding Copy On Write Arrays
* Introducing Queue and Deque, and Their Implementations

# Lambda Expressions

* Introduction
* Writing Lambda Expressions
* Functional Interfaces
* Types of Functional Interfaces
* Method reference

# Stream API

* Introduction
* Stream API with Collections
* Stream Operations

# JDBC

**Introduction to JDBC**

* Introduction to JDBC
* Architecture of JDBC
* Role of Driver Manager
* Understanding JDBC Driver Types

# Getting Started with JDBC

* Connecting to Database using JDBC
* Managing Database resources
* Handling JDBC Exceptions

# Performing Basic CRUD operations using JDBC

* Executing Static SQL Statements
* Iterating Through ResultSets
* Understanding ScrollableResultSets
* Understanding UpdatableResultSets
* Understanding PreparedStatement
* Retrieving Data Using PreparedStatement
* Inserting the Record
* Updating the Record
* Removing the Record

# J Unit Testing

Duration – 2 days

|  |
| --- |
| * Unit testing introduction |
| * Integration testing introduction Junit Intro |
| * Environment setup Writing tests |
| * API |
| * Assertions |
| * Junit Test Types Suite test Ignore test Time test Exceptions Test |
| * Parameterized Test TDD |
| * Developing code using TDD |