Course Design Document

Course Code	
Course Name	Core Java

Duration (in days)	7	Proficiency Level	Fundamental
Pre-requisites	• C Programming • Object Oriented Concepts	Target Audience	Campus Hires

Learning Outcome

At the end of the program, participants will learn

- Java Language Fundamentals
- Object Oriented Programming using Java
- Java Fundamentals: The Core Platform
- Exception Handling
- Functional Programming Fundamentals
- Generics
- Collection Framework
- Multithreading
- File IO
- Reflection and Annotation

Day-wise Session Plan

Day	Unit	Objectives	Hours
1	Java Language Fundamentals	Java, JRE, JDK, JVM Concepts Creating a Simple Application Variables Data Types • String Class • String Equality • String Representation of Non-string Values • Creating Jar Files (Command Prompt and Eclipse with manifest) Operators Conditional and Iteration Logic Arrays • Concepts • Arrays • For-each Loop	2

1	Java Language Fundamentals	Working with Packages • Introduction • What Is a Package? • Packages Create a Namespace • Determining a Type's Package • Packages Provide Access Boundaries Representing Complex Types with Classes • Introduction • Classes • Using Classes • Encapsulation and Access Modifiers • Method Basics • Exiting from a Method • Method Return Values • Static Members • Static Initialization Blocks • Field Encapsulation, Accessors, and Mutators • Overloading Class Initializers and Constructors • Introduction • Establishing Initial State • Field Initial State and Initializers • Constructor and Adding Multiple Constructors (Overloading) • Chaining Constructors and Constructor Visibility • Initialization Blocks • Initialization and Construction Order	6

2	Object Oriented Programming using Java	Class Inheritance • Introduction • Inheritance Basics and Typed References • Member Hiding and Overriding • Object Class • Equality • Special Reference: Super • Using Final and Abstract • Inheritance and Constructors Creating Abstract Relationships with Interfaces • Introduction • Introducing Interfaces & Implementing an Interface • Implementing a Generic Interface • Implementing Multiple Interfaces • Declaring an Interface	6
2	Java Fundamentals: The Core Platform	More About Data Types • Introduction • StringBuilder Class • Primitive Wrapper Classes and Type Conversions • Using Primitive Wrapper Classes • Wrapper Class Equality • Final Fields and Enumeration Types Inner Class Types, and Anonymous Classes • Introduction • Inner Classes • Anonymous Classes	2
3	Java Fundamentals: The Core Platform	String Formatting • Introduction • More Powerful Solutions to Creating String Representations	4

		• Joining Sequences of Values with StringJoiner • StringJoiner Edge Case Handling • Constructing String with Format Specifiers • Common Format Conversions Adding Type Metadata with Annotations • Introduction • The Need to Express Context and Intent • Using Annotations • Declaring Annotations • Accessing Annotations • Annotation Target and Retention • A Closer Look at Elements	
3	Java Fundamentals: The Core Platform	Exception Handling • Introduction • Error Handling with Exceptions • Handling Exceptions by Type • Exceptions and Methods • Throwing Exceptions and Custom Exceptions	6
4	Java Fundamentals: The Core Platform	Input and Output with Streams and Files • Introduction • Streams Overview • Reading and Writing with Streams • Common Stream Classes • Stream Errors and Cleanup • Chaining Streams • File and Buffered Streams • Accessing Files with the java.nio.file Package • Using Default File System and Zip File Systems • Demo: Creating a Zip File System • Demo: Copying and Writing Files to Zip File System Persisting Objects with Serialization • Introduction • Java Serialization Overview • Being Serializable • Serializing/Deserializing an Object • Class Version Incompatibility • Creating Class Version Compatibility • The Need for Custom Serialization • Customizing Serialization • Transient Fields • Example of files with Junit	6
5	Java Fundamentals: The Core Platform	Multi-Threading Basics • Introduction • A Quick Look at the Basics • The Move to Multithreading	6

		• Java Threading Foundation • Thread Pools • Creating a Closer Relationship Between Thread Tasks • Concurrency Issues • Coordinating Method Access • Manual Synchronization • Manually Synchronized Code • More Concurrency-related Types • Summary	
5	Java Collection Framework	What Are Collections and Why Use Them? • Introduction • The Problem with Arrays • Why You Should Use Collections Defining and Iterating Collections • Introduction • Collection of Collections • Collection Behaviors Collections with Iteration Order: Lists And Set • Introduction • Key Features • Coding with Lists	2
6	Java Collection Framework	Collection with Uniqueness: Sets • Introduction • Set Implementations • SortedSet and NavigableSet Sorting Techniques with Comparable and Comparator interface Collections with Modification Order: Queues, Deques, and Stacks • Introduction • Queues • Coding Queues • Priority Queues • Coding Priority Queues • Stacks and Deques • Coding Stacks and Deques Collections of Pairs: Maps • Introduction • Why Use a Map? • Views Over Maps • Sorted and Navigable Maps • Mutable HashMap Keys • TreeMap, LinkedHashMap, WeakHashMap, EnumMap • Summary	6
6	Java 8 Features	What's new in Interfaces • Static Methods • Default Methods • Functional Interfaces	2

7	Java 8 Features	Introducing Lambda Expressions • Motivation for Lambdas • Lambda Expression Overview • Lambda Expressions and Functional Interfaces • Using Lambda Expressions • Working with Method References The Stream API • Overview • Understanding the Stream API • Stream Processing • Collectors • Parallel Processing and Concurrency Date/Time API • Overview and Limitations of Previous API • The Date/Time API (JSR 310) • Creating and Working with LocalDate/LocalTime/LocalDateTime Instances • Formatting Date/Time • Localization / Time Zones • Periods and Durations • Performing Calculations on Data/Time types	8
		Total Hours	56