Mastering Java Programming

Target Audience :

- Individuals with a basic understanding of programming concepts.
- Students and recent graduates in computer science or related fields.
- Entry-level software developers looking to enhance their skills in Java.
- Professionals from other programming backgrounds who want to learn Java.
- Enthusiasts and hobbyists interested in developing Java applications.

Prerequisites:

- Familiarity with fundamental programming concepts is beneficial but not mandatory.
- A strong interest in learning Java and commitment to the course.

Content Coverage (Day Wise) :

Day 1

- Purpose of a Computer Program
- Translating High-Level Code to Machine Code
- Linked to Platform-Specific Libraries
- Platform-Dependent Programs
- Key Features of the Java Language
- Java Is Platform-Independent
- Java Programs Run In a Java Virtual Machine
- Procedural Programming Languages
- Java Is an Object-Oriented Language
- Topics
- Verifying the Java Development Environment
- Examining the Installed JDK (Windows Example): The Libraries
- LAB PRACTICE

Day 2

- Compiling and Running a Java Program
- Compiling a Program
- Executing (Testing) a Program
- Java Classes
- Program Structure
- Java Packages
- Using the Java Code Console
- Using the Java Code Console: Creating a New Java
- Using the Java Code Console: Creating a New Java Class
- LAB PRACTICE

Day 3

- The main Method
- A main Class Example
- Output to the Console
- Fixing Syntax Errors
- Creating a main Method
- Variables
- Variable Types
- Naming a Variable
- Uses of Variables
- Variable Declaration and Initialization
- String Concatenation
- String Concatenation Output
- Using String Variables
- LAB PRACTICE

Day 4

- String Concatenation

- String Concatenation Output
- Using String Variables
- int and double Values
- Initializing and Assigning Numeric Value
- Loops
- Processing a String Array
- Using break with Loops
- Using a Loop to Process an Array
- String Methods
- String classes
- LAB PRACTICE

Day 5

- Making Decisions
- The if/else Statement
- Boolean Expressions
- Relational Operators
- Examples
- Using if Statements
- What If There Are Multiple Items in the Shopping Cart?
- Introduction to Arrays
- Array Examples
- Array Indices and Length
- Declaring and Initializing an Array
- Accessing Array Elements
- LAB PRACTICE

Day 6

- Characteristics of Objects
- Classes and Instances
- The Customer Properties and Behaviors

- The Components of a Class
- Modeling Properties and Behaviors
- Customer Instances
- Object Instances and Instantiation Syntax
- The Dot (.) Operator
- Objects with Another Object as a Property
- Accessing Objects by Using a Reference
- Working with Object References
- References to Different Objects
- References and Objects in Memory
- Assigning a Reference to Another Reference
- Two References, One Object
- Arrays Are Objects
- Declaring, Instantiating, and Initializing Arrays
- LAB PRACTICE

Day 7

- String Class
- Concatenating Strings
- String Method Calls with Primitive Return Values
- String Method Calls with Object Return Values
- Java API Documentation
- Java Platform SE 8 Documentation
- Java Platform SE 8: Method Summary
- Java Platform SE 8: Method Detail
- indexOf Method Example
- StringBuilder Class
- StringBuilder Advantages over String for Concatenation (or Appending)
- StringBuilder: Declare and Instantiate
- StringBuilder Append
- Primitive Data Types

- Some New Integral Primitive Types
- Floating Point Primitive Types
- LAB PRACTICE

Day 8

- Basic Form of a Method
- Calling a Method from a Different Class
- Caller and Worker Methods
- A Constructor Method
- Writing and Calling a Constructor
- Calling a Method in the Same Class
- Method Arguments and Parameters
- Method Parameter Examples
- Method Return Types
- Method Return Types Examples
- Method Return Animation
- Passing Arguments and Returning Values
- More Examples
- Code Without Methods
- Better Code with Methods
- LAB PRACTICE

Day 9

- Static Methods and Variables
- Example: Setting the Size for a New Item
- Creating and Accessing Static Members
- When to Use Static Methods or Fields
- Some Rules About Static Fields and Methods
- Static Fields and Methods vs. Instance Fields and Methods
- Static Methods and Variables in the Java API
- Examining Static Variables in the JDK Libraries

- Using Static Variables and Methods: System.out.println
- More Static Fields and Methods in the Java API
- Converting Data Values
- Passing an Object Reference
- What If There Is a New Object?
- Passing by Value
- Reassigning the Reference
- Passing by Value
- Method Overloading
- Using Method Overloading
- Method Overloading and the Java API
- LAB PRACTICE

Day 10

- What Is Access Control?
- Access Modifiers
- Access from Another Class
- Another Example
- Using Access Control on Methods
- Encapsulation
- Get and Set Methods
- Why Use Setter and Getter Methods?
- Setter Method with Checking
- Using Setter and Getter Methods
- Initializing a Shirt Object
- Constructors
- Shirt Constructor with Arguments
- Default Constructor and Constructor with Args
- Overloading Constructors
- LAB PRACTICE

Day 11

- Relational Operators
- Testing Equality Between String variables
- Common Conditional Operators
- Ternary Conditional Operator
- Using the Ternary Operator
- Handling Complex Conditions with a Chained if Construct
- Determining the Number of Days in a Month
- Chaining if/else Constructs
- Handling Complex Conditions with a switch Statement
- Coding Complex Conditions: switch
- switch Statement Syntax
- When to Use switch Constructs
- Working with an IDE Debugger
- Debugger Basics
- Setting Breakpoints
- The Debug Toolbar
- Viewing Variables
- LAB PRACTICE

Day 12

- Displaying a Date
- Class Names and the Import Statement
- Working with Dates
- Working with Different Calendars
- Some Methods of LocalDate
- Formatting Dates
- Using the args Array in the main Method
- Converting String Arguments to Other Types
- Exercise 11-2: Parsing the args Array

- Describing Two-Dimensional Arrays
- Declaring a Two-Dimensional Array
- Instantiating a Two-Dimensional Array
- Initializing a Two-Dimensional Array
- Some New Types of Loops
- Repeating Behavior
- A while Loop Example
- Coding a while Loop
- while Loop with Counter
- LAB PRACTICE

Day 13

- ArrayList Class
- Benefits of the ArrayList Class
- Importing and Declaring an ArrayList
- Working with an ArrayList
- Implementing Inheritance
- More Inheritance Facts
- Inheritance
- Constructor Calls with Inheritance
- Inheritance and Overloaded Constructors
- More on Access Control
- Overriding Methods
- Overriding a Method: What Happens at Run Time?
- Polymorphism
- Superclass and Subclass Relationships
- Using the Superclass as a Reference
- Polymorphism Applied
- Accessing Methods Using a Superclass Reference
- Casting the Reference Type
- instanceof Operator

- Abstract Classes
- Extending Abstract Classes
- LAB PRACTICE

Day 14

- The Object Class
- Calling the toString Method
- Overriding toString in Your Classes
- The Multiple Inheritance Dilemma
- The Java Interface
- Multiple Hierarchies with Overlapping Requirements
- Using Interfaces in Your Application
- Implementing the Returnable Interface
- Access to Object Methods from Interface
- Casting an Interface Reference
- The Collections Framework
- ArrayList Example
- List Interface
- Using a Lambda Expression with replaceAll
- Lambda Expressions
- The Enhanced APIs That Use Lambda
- Lambda Types
- LAB PRACTICE

Day 15

- What Are Exceptions?
- Examples of Exceptions
- Code Example
- Another Example
- Types of Throwable classes
- Error Example: OutOfMemoryError

- Normal Program Execution: The Call Stack
- How Exceptions Are Thrown
- Working with Exceptions in NetBeans
- The try/catch Block
- Program Flow When an Exception Is Caught
- When an Exception Is Thrown
- Throwing Throwable Objects
- Uncaught Exception
- Exception Printed to Console
- Calling a Method That Throws an Exception
- Working with a Checked Exception
- Deploying and Maintaining the Application
- LAB PRACTICE