Mphasis
The Next Applied

Basics of Java

Session Plan

Core Java

- ✓ Introduction
- ✓ Wrapper Classes
- ✓ Object Oriented Programming
- ✓ Arrays API
- ✓ String Classes
- ✓ Working with Exceptions
- ✓ Design Patterns
- ✓ Java 8 Features
- ✓ Working with the Date/Time API

Session Plan

- ✓ Generic Classes
- ✓ Collections Framework
- ✓ Working with Stacks
- ✓ Working with Queues
- ✓ Sorting & Searching Algorithms
- ✓ Input & Output Streams
- ✓ Multi-Threading
- ✓ JDBC API

Session Plan

JUnit

- ✓ Performing Unit Testing using JUnit4
- ✓ Deployment and Application Enhancement
- ✓ Memory Management

Maven

- ✓ Maven Repositories & Dependency Management
- ✓ Running Tests & Generating Reports

Day 1

Basics of Java

Introduction

- Introduction to Java
- JVM Architecture
- Installation of Java
- Configuring SDE Eclipse
- Understanding how JRE and JVM works

Basics

- Class Structure
- Java Keywords
- Primitive data types
- Creating primitive variables
- Using operators
- Using if-else and switch statements
- Iterating with loops: while, do-while, for, enhanced for
- Wrapper Classes and Autoboxing concepts

Day 2

Object Oriented Programming

Object Oriented Programming

- Working on Constructors
- Achieving Encapsulation
- Code Reusability via Inheritance
- Achieving Polymorphism
- Working on methods of java.lang.Object class
- Object Casting
- Passing Objects as Arguments
- Abstraction via Abstract Classes and Interfaces
- Diamond Problem using Interfaces.
- Creating Static Classes and Static Methods

Day 3

Arrays API, String Classes

Arrays API and String Classes

- Single-Dimensional Array
- Multi-Dimensional Arrays
- Array of Objects
- Arrays utility class
- String Class
- StringBuffer class
- StringBuilder class
- Introduction to Regex (Regular Expression)

Day 4 Working with Exceptions

Working with Exceptions

- Defining the purpose of Java exceptions
- Using the try and throw Statements.
- Using the catch, multi-catch, and finally clauses
- Autoclose resources with a try-with-resources statement
- Recognizing common exception classes and categories
- Creating custom exceptions

Day 5

Design Patterns

Design Patterns

- Singleton Design Pattern
- Factory Design Pattern
- Abstract Factory Design Pattern
- Builder Design Pattern
- Template Method Design Pattern
- Bridge Design Pattern
- Proxy Design Pattern
- Creating Immutable classes

Day 6

Java 8 Features and Generics

Java 8 Features and Generics

- Motivation for Lambdas
- Lambda Expression Overview
- Lambda Expressions and Functional Interfaces
- Method References
- The Date/Time API (JSR 310)
- Use of LocalDate/LocalTime/LocalDateTime Instances
- Dates and Times across Time Zones
- Formatting Dates
- Inheritance with Generic Types
- Wildcard Parameter Types (bounded & unbounded)

Day 7 Collections Framework

Collections Framework

- Collections Overview
- Using the type inference diamond to create an object
- Creating a collection by using generics
- Implementing an ArrayList, LinkedList, Vector
- Implementing a HashSet, TreeSet
- Implementing a HashMap, TreeMap, HashTable
- Ordering collections – Comparable & Comparator
- Utility Classes - Collections and Arrays

Collections Framework

- Stream API
- java.util.function Package – Predicate, Consumer, Function, and Supplier
- Stream Operations
- Stream map method
- FindFirst and Lazy Operations
- Sorting a Stream

Day 8

Stacks & Queues

Working with Stacks

- Introducing the Stack
- Stack Using Arrays - Is Empty, Is Full, and Size
- Stack Using Arrays - Push
- Stack Using Arrays - Pop
- Stack Using Arrays - Peek
- Stack Using Linked Lists - Push
- Stack Using Linked Lists - Pop and Peek

Working with Queues

- Introducing the Queue
- Queue Using Arrays - Is Full, Is Empty, and Size
- Queue Using Arrays - Enqueue, Dequeue
- Queue Using Arrays - O(N) Enqueue and Peek
- Circular Queue - Is Full, Is Empty, and Enqueue
- Circular Queue - Dequeue and Peek
- Queue Using Linked Lists - Enqueue, Dequeue and Peek

Day 9 Sorting & Searching Algorithms

Sorting & Searching Algorithms

- Sorting Algorithms and Trade-offs
- Implementing Selection Sort
- Implementing Bubble Sort
- Implementing Insertion Sort
- Implementing Merge Sort
- Implementing Quick Sort
- Implementing Linear Search
- Implementing Binary Search

Day 10

Input/Output Streams, Multi-Threading

Input/Output Streams

- Describing the basics of input and output in Java
- Read and write data from the console.
- Using streams to read and write files.
- Writing and read objects using Serializable.

Multi-Threading

- Describing operating system task scheduling
- Creating worker threads using Runnable and Callable
- Thread Life Cycle

Day 11

Multi-Threading

JDBC API

Multi-Threading

- Synchronization in Threads
- Inter-Thread Communication
- Avoiding common multithreading pitfalls
- Schedulers, Timers

JDBC API

- Defining the layout of the JDBC API
- Connecting to a database by using a JDBC driver
- Submitting queries and get results from the Database.
- Specifying JDBC driver information externally

Day 12

JDBC API, JUnit4

JDBC API

- Performing CRUD operations using the JDBC API – Statement & PreparedStatement
- Metadata using ResultSetMetaData and DatabaseMetaData

Performing Unit Testing using JUnit4

- Overview
- Tests, Assertions, and Fixtures
- Writing and Running Tests
- Assertions
- Test Fixtures, @Before and @After, @BeforeClass and @AfterClass

Day 13

Deployment,
Memory Management

Performing Unit Testing using JUnit4

- Test cases for Exception and Timeout
- Parameterized Tests
- Test Suites

Deployment and Application Enhancement

- Packages
- Creating a JAR File
- Client/Server Architecture
- Running a JAR File from the Command Line

Memory Management

- Garbage Collection API
- Make an object eligible for GC.
- Requesting JVM to run Garbage Collector

Day 14 Apache Maven

Memory Management

- How and when to use Finalization
- Types of JVM Garbage Collectors

Maven Repositories & Dependency Management

- Setting up Maven
- Navigating a Project Structure
- The POM File
- Building, Testing, and Packaging a Project
- Overview of Dependency Management and Repository
- Maven Lifecycles and Phases
- Configuring and Using Plugins
- Developing a Basic Plugin
- Built in Archetypes
- Generating a Web Project
- Maven Build Profiles
- Working with Build Profiles

Running Tests & Generating Reports

- Overview of Testing
- Adding Test-scoped Dependencies
- Running Tests
- Generating Test Reports
- Using the Site Lifecycle
- Customized Site Configuration
- Using the Javadoc Plugin
- Integrating Maven with Eclipse

Mphasis
The Next Applied

THANK YOU

About Mphasis

Mphasis (BSE: 526299; NSE: MPHASIS) applies next-generation technology to help enterprises transform businesses globally. Customer centricity is foundational to Mphasis and is reflected in the Mphasis' Front2Back™ Transformation approach. Front2Back™ uses the exponential power of cloud and cognitive to provide hyper-personalized (C=X2C^{2} $_{TM}$=1) digital experience to clients and their end customers. Mphasis' Service Transformation approach helps 'shrink the core' through the application of digital technologies across legacy environments within an enterprise, enabling businesses to stay ahead in a changing world. Mphasis' core reference architectures and tools, speed and innovation with domain expertise and specialization are key to building strong relationships with marquee clients. Click here to know

Important Confidentiality Notice

This document is the property of, and is proprietary to Mphasis, and identified as "Confidential". Those parties to whom it is distributed shall exercise the same degree of custody and care afforded their own such information. It is not to be disclosed, in whole or in part to any third parties, without the express written authorization of Mphasis. It is not to be duplicated or used, in whole or in part, for any purpose other than the evaluation of, and response to, Mphasis' proposal or bid, or the performance and execution of a contract awarded to Mphasis. This document will be returned to Mphasis upon request.

Any Questions?

FOLLOW US IN THE LINK BELOW

@Mphasis