Lara Technologies

A Unique future creator in Java

Let us create world not Ad

 # 152, 13th Main Road, 1st cross,
BTM Layout, Nearby Udupi
garden signal, 1st Stage, Aicobo Nagar,
Bangalore - 560 029,
Ph: +91- 9972006654,
Website: www.laratechnology.com
Mail id : info@laratechnology.com

Core Java Advance

- ➢ Enums
- ➢ Static imports
- ➢ Inner classes
- ➢ Annotations.
- ➢ Wrapper classes

Exception Handling:

- ❖ Different types of abnormal conditions.
- ❖ Why do we require Exception handling
- ❖ try/ catch/ finally
- ❖ Deviations to finally block.
- ❖ Return statement inside try/catch/finally
- ❖ Errors and Exceptions
- ❖ Checked and Unchecked
- ❖ Explanation of Error types
  - ➢ StackOverFlowError
  - ➢ OutOfMemoryError
  - ➢ NoClassDefinationFoundError
  - ➢ NoSuchMethodError etc…
- ❖ Explanation of UncheckedException types
  - ➢ ArithmaticException
  - ➢ NumberFormatException
  - ➢ NullPointerException
  - ➢ ArrayIndexOutOfBoundException
  - ➢ ClassCastException etc…

- Explanation of CheckedException types
  - SQLException
  - ClassNotFoundException
  - IOException
  - FileNotFoundException
  - CloneNotSupportedException
  - ParseException
  - InterruptedException etc…
- Explanation of unreachable statements.
- throws keyword and its importance in unchecked.
- Rules of method overriding in case of throws.
- Order of catch blocks
- throw keyword and its usage
- Developing user defined exception class.
- Differentiate throws and throw.

Assertions

- Why do we require assertions.
- enabling and disabling the assert statements in execution wise, package wise and class wise.
- Types of assert statements.
- Usage of assert as an identifier in older versions of JDK
- In appropriate usage of assert statements
- Difference between throw and assert.
- Enable/Disable assert statements in Eclipse

Object Class

- ❖ toString()
- ❖ hashCode()
- ❖ equals()
- ❖ finalize()
- ❖ Garbage collector.
- ❖ clone()
    - ➢ Deep copy / Shallow copy
- ❖ getClass()
- ❖ java.lang.Class methods
    - ➢ getFields()
    - ➢ getDeclaredFields()
    - ➢ getMethods()
    - ➢ GetDeclaredmethods()
- ❖ Reflection API

Multi Threading.

- ❖ Multi Tasking
- ❖ Multi Processing.
- ❖ Multi threading.
- ❖ Types of Threads (user and daemon)
- ❖ Thread with Runnable Interface
- ❖ Thread with Thread class
- ❖ Developing threads with inner classes.
- ❖ Default properties of threads.
- ❖ Finding current thread.
- ❖ Thread unique id.

- Thread name
- Thread priority.
- Thread daemon status.
- Threads join.
- Thread sleep.
- Thread interruptions.
- Synchronization
    - Synchronization methods
    - Synchronization blocks
- Dead Lock
- Inter Thread communication
    - wait()
    - notify()
    - notifyAll()
- Thread pool
- Thread group
- Thread life cycle
- Thread yield.
- Thread Locale
- Difference between Thread and Runnable.

Strings

- ❖ String class basic information.
- ❖ Some important methods of String class.
  - ➢ toString()
  - ➢ hashCode()
  - ➢ equals()
  - ➢ length()
  - ➢ concat()
  - ➢ trim()
  - ➢ charAt()
  - ➢ indexOf()
  - ➢ lastIndexOf()
  - ➢ substring()
  - ➢ split()
  - ➢ toUpperCase()
  - ➢ toLowerCase()
  - ➢ equalsIgnoreCase()
  - ➢ startsWith()
  - ➢ endsWith()
  - ➢ replace()
  - ➢ replaceAll()
  - ➢ "==" operator
  - ➢ size()
  - ➢ Differentiate equals and == operator etc..
  - ➢ Mutability
  - ➢ Capacity

- ➢ Buffer
- ➢ Thread safeness.
- ➢ Extra methods like append(), reverse(), delete(), etc
- ➢ Flags, width, precision, conversion chars.
- ➢ Patterns, Matcher
- ➢ \d, \d+, \s, \w and so on
- ➢ + operator

❖ Explanation String memory management
❖ StringBuffer class
❖ Differentiate String class and StringBuffer class
❖ StringBuilder class
❖ Differentiate StringBuffer and StringBuilder.
❖ Formatters
❖ Regular Expression
❖ StringTokenizer.
❖ Date, Calendar
❖ NumberFormat and DateFormat Local.

Arrays:

❖ Declaration, Definition, Initialization
❖ One dimensional Array
❖ Multi dimensional Array
❖ Java.util.Arrays
❖ Comparable Interface
❖ Comparator Interface

Collection API:

- ❖ Limitations of arrays.
- ❖ Introduction to Collection API
- ❖ Introduction of Java.util package
- ❖ Different streams of collection API.

List stream

- ❖ List overview
- ❖ Important members from List stream
  - ➢ ArrayList
  - ➢ LinkedList
  - ➢ Vector

- ❖ How to read elements from Collection Object
  - ➢ Through Regular for loop
  - ➢ Through Enhanced for loop
  - ➢ Through toString().
  - ➢ Through Iterator
  - ➢ Through ListIterator

- ❖ Experimenting all basic operations of Collection objects
- ❖ Sorting List elements by using Comparable.
- ❖ Sorting List elements by Comparator.
- ❖ Experimenting Collections utility class.
- ❖ Developing our own Stack by using Linked List
- ❖ Developing our own Queue by using Linked List
- ❖ Developing our own ArrayList class
- ❖ Developing Stack and Queue without using Collection classes.

- Development of different types of LinkedList classes
  - Single
  - Double
  - Circular
- Difference between ArrayList and LinkedList
- Enumeration

Queue stream

- Queue overview
- PriorityQueue
- Usage of Comparator and Comparable in Queue stream.
- BlockingQueue.
- ArrayBlockingQueue
- DelayQueue
- BlockingDeque
- LinkedBlockingDeque

Set stream

- Set overview
- Set uniqueness
- Usage of hashCode() and equals() methods of Object class.
- Hash Bucketing.
- Important members of Set stream
  - HashSet
  - LinkedHashSet
  - TreeSet
  - NavigableSet
- Usage of Comparable and Comparator interfaces for TreeSet

Map stream

- ❖ Map overview.
- ❖ Important members of Map stream
  - ➢ HashMap
  - ➢ HashTable
  - ➢ Properties
  - ➢ TreeMap
  - ➢ NavigableMap
  - ➢ LinkedHashMap
  - ➢ ConcurrentMap

- ❖ Usage of Comparable and Comparator interfaces for TreeMap.
- ❖ Developing our own ThreadLocal
- ❖ Developing Object Pool Design pattern
- ❖ Synchronization, developing our own synchronized collections.
- ❖ Fail fast and Fail Safe
- ❖ Concurrent package

Generics:

- ➢ Why Use Generics?
- ➢ Generic Types
- ➢ Raw Types
- ➢ Generic Methods
- ➢ Bounded Type Parameters
- ➢ Generic Methods and Bounded Type Parameters
- ➢ Generics, Inheritance, and Subtypes

- ➢ Type Inference
- ➢ Wildcards
- ➢ Upper Bounded Wildcards
- ➢ Unbounded Wildcards
- ➢ Lower Bounded Wildcards
- ➢ Wildcards and Subtyping
- ➢ Wildcard Capture and Helper Methods
- ➢ Guidelines for Wildcard Use
- ➢ Type Erasure
- ➢ Erasure of Generic Types
- ➢ Erasure of Generic Methods
- ➢ Effects of Type Erasure and Bridge Methods
- ➢ Non-Reifiable Types
- ➢ Restrictions on Generics

File handling

- ❖ File
- ❖ FileReader
- ❖ FileWriter
- ❖ BufferedReader
- ❖ BufferedWriter
- ❖ BufferedInputStream
- ❖ BufferedOutputStream
- ❖ Serialization
- ❖ Deserialization
- ❖ transient key word.

- Externalization
- DeExternalization
- Java.io.Console
- Customizing S.O.P
- PrintStream