Lara Technologies

A Unique future creator in Java

Let us create world not Ad

 # 8, 2nd Floor, BTM Layout,
2nd Stage, Bangalore - 560 076,
Ph: 080 - 41680090, (+91) 9663646876
Website: www.laratechnology.com

Mail: info@laratechnolgy.com

New Batches

1. **Class Room Classes for Java, J2EE and Frameworks**

   Start Date   :   **July 10th**

   Timing   :   **7.00PM to 9.00 PM**

   Duration   :   **5 month**

   (Every day classes including weekends)

2. **Class Room Classes for Java, J2EE and Frameworks**

   Start Date   :   **June 19th**

   Timing   :   **4.30PM to 6.30 PM**

   Duration   :   **5 month**

   (Every day classes including weekends)

   Fee

   Complete Package :   **Rs. 24,000**

   Only core Java   :   **Rs. 12,000**

   Only J2EE   :   **Rs. 12,000**

   Only Frameworks   :   **Rs. 12,000**

After 5 months of training you will be able to apply to the following JOBS.

1.	**Pure core Java requirements. (Core Java, SQL and JDBC)**
2.	**Web Application requirements. (Core Java, SQL and JDBC , Servlets, and JSP)**
3.	**Enterprise Application Requiremennts(Core Java, SQL and JDBC, Servlets, JSP, Hibernate, Spring and Struts)**
4.	**Database developer (SQL)**
5.	**Web Designer (HTML and CSS)**

6.	**Web Developer (HTML, CSS, Javascript, JQuery and Angular JS)**

1. **Few words about Lara**

   Mr. Ramesh, a founder and director of Lara Technologies worked in Top MNC companies like IBM, GE and Wipro for almost 9 years. With this rich industry experience, He started Lara in 2005. In these 12 years of training, He trained more than 50,000 people so far. More than 90% of trained students placed in various Top MNC companies. Every company knows "Lara Technologies" as a best and only training center for Java, J2EE and Frameworks.

2. **Main course content**

   - Core Java
   - Logical coding (number system, patterns printing, data strutctures, and algos)
   - SQL
   - JDBC
   - Web designing (HTML, HTML5, CSS and CSS3)
   - Web Programing (Java Script, JQuery, JSON, AJAX and Angular JS).
   - ANT, Maven, SVN, JUnit, Log4J and Eclipse.
   - Servlets
   - JSP

- ➢ Design Patterns (Core Java and Web application patterns)
- ➢ XML
- ➢ WebServices
- ➢ Struts2.0
- ➢ Hibernate
- ➢ Spring

(Detailed course content given at the end of the brochure)

3. **Why you have to choose Lara to learn Java, J2EE and Frameworks**

   We have very rich industry experience. Our course content is very suitable to every company requirement. We had completed almost 150 batches so far. Batch by batch, content and examples are increasing to make the training very effective. We are focusing every concept which is required for interviews and for project development. We are already proved that, "the best in India". Our Laraites only best employees in any project development. Finally 100% Job Guarantee.

4. **What is the teaching style in Lara**

   Every concept is with full of examples. Without example, no concept is driving. Concept by concept with a clear link, even non computer student also can understand very easily.

5. **Lab facility in Lara?**

   Complete one floor is allocated for lab. You can bring your lappy and practice in the lab. If you don't have lappy, then you can use our desktops. You can work as much time as you wish. There will be instructors to help you while you are in struck.

6. **Evaluations/Tests in Lara**

   After every topic, there will be one exam. Immediately after exam, key will be released. Within 2 to 3 days results will be announced.

7. **Materials and exercises (For essential topics).**

There are two types of materials. One is Lab manual, where you can find only examples for all the concepts. 2nd one is FAQS, where you can find questions and answers, which are asking in the interviews. For every topic, exercise will be provided at the end. Material will be provided for every topic separately in the class room itself. If you miss on the day of material distribution, then you have to take photo copy by taking from your friends.

8. **Logical coding, algorithms and Data Structures?**

We are covering all logical coding and algorithms, which are very much required for the interviews. We are focusing on,

1. Number system
2. Patterns related (Square, Triangles, Pyramid and Dimond)
3. Strings
4. String and FileHandling
5. Data Structures (arrays, stack, queue, linked lists and tree)
6. Recursive algorithm.
7. Sort and Search Algorithms.

9. **Projects in the course?**

Prototype development in various skills,

After SQL, we will design the database for one case study with all table relations, like

one-to-one, one-to-many, many-to-many. Then we will develop a CRUD (create, read, update and delete) in various skill combinations.

1. Core Java and JDBC
2. Core Java and Hibernate
3. Servlets/JSP and JDBC
4. Servlets/JSP and Hibernate

5. Struts2.x and JDBC

6. Struts2.x and Hibernate

7. Spring DAO and JDBC

8. Spring ORM and Hibernate.

9. Spring MVC and ORM.

10. **Placement guarantee**

100% Job guarantee. After 2 and half months training, every student is allowed to attend the drives. A student can attend any no of drives till he/she get placed. If you are following regularly all the concepts in the training, then 100% you will be getting placed from us itself. We are interacting with more than 500 companies. Every month, new companies are coming up.

Recent visited companies list provided at the end of this brochure.

11. **Is there any pass out will be major criteria in the placements?**

Absolutely no. There are several companies, which are considering pnly knowledge of Java rather than year of pass out. We placed even 2010 passed out also as a fresher.

12. **Is there any educational percentages will be major criteria in the placements?**

Absolutely no. There are several companies, which are considering only knowledge in Java rather than percentages in the education. We have placed the students whose percentages are less than 60% in entire education

13. **What could be the package while placing through "Lara placement"?**

It varies from company to company. We are ensuring minimum 15K/month. But majority of the companies are paying minimum 2.5 Lakhs/annum to maximum even 6.00 Lakhs/annum.

14.   OCJP (SCJP)

Our core java syllabus is very close to OCJP certification. No need of additional preparation. Immediately after core java training, you can attempt OCJP. Definitely you can score more than 90%. All OCJP previous exam questions are covered in our lab manuals and exercises.

15.   How many students might be in one batch?

Its depending on time of the batch. Roughly 30 to 40 students in one batch. But you don't require worrying about this. No student will get even a single doubt. Our delivery will be with full of examples and chain of dependencies. Same concept will be repeating again and again with various examples.

16.   Is there any chance of completing the course in the short period?

No. Not possible. Because of content is more and unable to cover in the short period. % months itself shortest time.

17.   What kind of benefits to Fresher's?

1. Main programing stream like Core Java, SQL, J2EE and frameworks
2. Logical coding which is very much required to fresher job interviews.
3. Aptitude and Reasoning which are very much essential to fresher written tests.
4. Mock interviews, which are very much essential to the fresher as they might have a fear on attending the interviews.

18.   What kind of benefits to working professionals?

1. Main programing stream like Core Java, SQL, J2EE and frameworks.

---

2. More practical exposure as every concept is covering with more examples.

3. Prototype developments in various skills will be helping to involve in any project immediately.

19. **Academic projects to pursuing students.**

We have a separate faculty for academic projects. He will be developing and helping academic projects for all pursuing students. All projects are DB interaction projects. Our faculty will help you to develop the project on your own. We will not be providing source code. You have to develop the code with our guidance. You can use any skills like servlets/jsp or Spring/Hibernate. No separate admission only to academic projects. Whoever joining the main course will be allowed to involve in the academic project development. No extra fee.

20. **Course Completion Certificate (CCC)**

After completing the course, you have to undergo a written test in entire course syllabus and should secure minimum of 50% to get CCC. We conduct CCC test on every weekend. If you fail to score minimum of 50%, then prepare well and write once again after 2 weeks.

21. **Can non computer student can learn Java, J2EE and Frameworks from Lara?**

Yes absolutely. Minimum graduation knowledge is highly sufficient to understand. Our course is designed such a way like any graduate can understand. You don't require any pre computer knowledge. No need of knowing C (or) C++ before learning Java.

22. **Is it mandatory to write Lara exams by working students?**

Yes. Whichever exams conducting in the convenient time of working professionals are mandatory to write.

Core Java Part 1

1. JDK
   Downloading, installation and hello world development.
2. PATH environment Variable
3. –d option of javac command.
4. –cp option of java command.
5. Key words, literals, identifiers and buil-in classes.,
6. println and print
7. Local Variable Introduction
8. unary operators
   - increment(pre & post)
   - decrement(pre & post)
9. Conditiona statements (if, else, switch)
10. Iterations (for, while and do-while loops)
11. Methods
12. Introduction of global variables
13. Introduction of SIB
14. Multiple classes
15. Introduction to object
16. Pass by reference
17. Constructors
18. Introduction of IIB
19. Encapsulation
20. Inheritance
21. Combination of Constructor, this(), super(), SIB, IIB, inheritance
22. Packages
23. Access levels
24. Abstract class
25. Interfaces
26. Combination of interfaces and abstract classes
27. Method overloading
28. Method overriding
29. Usage of super and this statements inside the methods
30. Casting
31. Polymorphism
32. Static members are not involving in the polymorphism.
33. final keyword
34. this key word.
35. CLASSPATH
36. javap, javadoc, jar commands.
37. Eclipse
38. Enhanced for loop
39. Var-args
40. Command line arguments
41. Scanner class

Core Java Part-2

- Creational Design patterns
- Enums
- Static imports
- Inner classes
- Annotations.
- Wrapper classes

1. **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 uncheked
- Explanation of Error types
  - StackOverFlowError
  - OutOfMemoryError
  - NoClassDefinationFoundError
  - NoSuchMethodError etc…

- ❖ Explanation of UncheckedException types
  - ➢ ArithmaticException
  - ➢ NumberFormatException
  - ➢ NullPointerException
  - ➢ ArrayIndexOutOfBoundExcep tion
  - ➢ ClassCastException etc…
- ❖ Explanation of CheckedException types
  - ➢ SQLException
  - ➢ ClassNotFoundException
  - ➢ IOException
  - ➢ FileNotFoundException
  - ➢ CloneNotSupportedExceptio n
  - ➢ 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.

2. **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

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

4. **Multi Threading.**
- ❖ Multi Tasking
- ❖ Multi Processing.
- ❖ Multi threading.
- ❖ Types of Threads (user and deamon)
- ❖ 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 Runnasble.

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

- ❖ Explanation String memory management
- ❖ StringBuffer class
  - ➢ Mulatability
  - ➢ Capacity
  - ➢ Buffer
  - ➢ Thread safeness.
  - ➢ Extra methods like append(), reverse(), delete(), etc
- ❖ Differentiate String class and stringBuffer class
- ❖ StringBuilder class
- ❖ Differentiate StringBuffer and stringBuilder.
- ❖ Formatters
  - ➢ Flags, width, precision, conversion chars.
- ❖ Regular Expression
  - ➢ Patterns, Matcher
  - ➢ \d, \d+, \s, \w and so on
- ❖ StringTokenizer.
- ❖ Date, Calander
- ❖ NumberFormat and DateFormat Locale.

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

7. **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

8. **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

9. **File handling**
   - ❖ File
   - ❖ FileReader
   - ❖ FileWriter
   - ❖ BufferedReader
   - ❖ BufferedWriter
   - ❖ BufferedInputStream
   - ❖ BufferedOutputStream
   - ❖ Serialization

- ❖ Deserialization
- ❖ transient key word.
- ❖ Externalization
- ❖ DeExternalization
- ❖ Java.io.Console
- ❖ Customizing S.O.P
- ❖ PrintStream

10. **JVM architecture**
    - ❖ Class loader subsystem.
    - ❖ Runtime data area
      - ➤ Method area
      - ➤ Heap
      - ➤ Java stacks
      - ➤ PC registers
      - ➤ Native method stacks
    - ❖ Execution engine.

11. **Design Patterns**
    - ❖ Structural Design Patterns
    - ❖ Behavioral Design Patterns

SQL

- ❖ Downloading and installation of Oracle DataBase 11g Express Edition
- ❖ Downloading and installation of SQL developer
- ❖ Sql commands for table, sequence and procedures
  - ➤ DDL
  - ➤ DML
  - ➤ DQL
- ❖ SQL Functions
- ❖ SQL without Constraints
- ❖ SQL with Constraints
  - ➤ Not Null
  - ➤ Unique
  - ➤ Combination of not null and Unique
  - ➤ Primary key
  - ➤ Composite Key
    - • Composite Unique Key

- Composite Primary Key
- Foreign Key
❖ Foreign key relation ship
  ➢ One to-one-relationship
  ➢ One to-many-relationship / Many-to-one relationship
  ➢ Many-To-Many relationship
❖ SQL JOINS
  ▪ INNER-JOIN:
  ▪ LEFT-OUTER-JOIN
  ▪ RIGHT-OUTER-JOIN
  ▪ FULL-OUTER-JOIN
    In one-to-one, one-to-many and many-to-many mappings
❖ Date, TimeStamp, RowId, RowNum

JDBC

❖ JDBC basics
❖ Explanation of Drivers
❖ Through Type 1 Driver and Type 4 Driver
  ➢ DataBase connection with Statement Object
  ➢ ResulSet
  ➢ ResulSetMetadata
  ➢ DataBaseMetaData
  ➢ Prepared Statement Object to DataBase connection
  ➢ Stored Procedures and Callable Statement Object to DataBase connection
    - IN
    - OUT
    - IN OUT
    ▪ Transaction Usage
❖ JDBC 2.0 features

➢ ResultSetscrollability
➢ ResultSet Updatability.
➢ Batch Updates
➢ Storing and reading Image/audio/vedio file from the database
❖ JDBC 3.0 features
  ➢ Retrieval of auto generated keys.
  ➢ ParameterMetaData.
  ➢ SavePoint
❖ JDBC 4.0 features
  ➢ ROWID data type
  ➢ Auto loading of JDBC driver
  ➢ SQLException enhancement.
❖ Interacting with Excel Sheet by using JDBC
❖ Interacting with Excel sheet by using POI API.
❖ Exploring different usages of Date and Timestamp data types

Hibernate

❖ Differences between JDBC and Hibernate
❖ Downloading Hibernate and setting up for development.
❖ POJO class Development, Hibernate configuration file development, Hibernate mapping file development
❖ Object Relational mapping introduction
❖ simple operations
  ➢ save(), persist(), update(), delete(), saveOrUpdate(), get(), load(),
❖ Lazy loading and Eager loading
❖ Criteria API

- ➢ Different methods in Restrictions,
- ➢ Every where clause achievement.
- ➢ Different ways of Criterion usage.
- ➢ Index Parameters, Orders, Pagination,
- ❖ Query API
  - ➢ Through HQL
  - ➢ Through Native SQL
  - ➢ Through Named SQL
  where clause achievement.
  Index Parameters, Orders, Pagination,
- ❖ Different types of Generators (native, increment, sequence, assigned)
- ❖ Annotation Mapping
  - ➢ For all basic operations (save(), persist(), update(), delete(), saveOrUpdate(), get(), load())
  - ➢ For Generators
  - ➢ For Named Queries
- ❖ Component mapping (XML and Annotation Mapping)
- ❖ Composite Primary Key (XML and Annotation Mapping)
- ❖ Polymorphic mappings (XML and Annotation Mapping)
- ❖ Collection Mappings (XML and Annotation Mappings)
- ❖ Using Maven tool for Hibernate development.
- ❖ Association mappings and with XML and Annotation mappings
  - ➢ One-to-one
  - ➢ One-to-many
  - ➢ Many-to-many
- ❖ Cache management

- ➢ 1st level and 2nd level
- ❖ Multiple Configuration files connection
- ❖ Interacting with multiple Databases (Oracle and MySql)
- ❖ Programmatic Configuration
- ❖ Thread wise Session management. (getCurrentSession())
- ❖ Using Hibernate In Servlets/JSP application to develop CRUD
- ❖ Using Hibernate in Struts based application to develop CRUD
- ❖ Using Hibernate in Spring MVC based application to develop CRUD.

HTML

- ❖ HTML introduction
- ❖ HTML editors
- ❖ Html Basics
- ❖ HTml elements
- ❖ Html Attributes
- ❖ Html Headings & paragraphs
- ❖ Html Styles
- ❖ Html Formatting
- ❖ Html comments
- ❖ Html colors
- ❖ Html CSS
- ❖ Html links
- ❖ Html images
- ❖ Html tables
- ❖ Html lists
- ❖ Html blocks
- ❖ Html classes
- ❖ Html I frames
- ❖ Html file paths
- ❖ Html head
- ❖ Html layout

- ❖ Html forms
- ❖ Html form elements
- ❖ Html input types
- ❖ Html input attributes
- ❖ html Methods

HTML5
- ❖ Html5 intro
- ❖ Html5 elements
- ❖ Html5 semantics
- ❖ Html5 style guide
- ❖ Html5 validations & components

CSS
- ❖ CSS Introduction
- ❖ CSS Syntax
- ❖ CSS Colors
- ❖ CSS Backgrounds
- ❖ CSS borders
- ❖ CSS margins
- ❖ CSS paddings
- ❖ CSS height/ width
- ❖ CSS text
- ❖ CSS fonts
- ❖ CSS lists
- ❖ CSS tables
- ❖ CSS display
- ❖ CSS inline - block
- ❖ CSS align
- ❖ CSS dropdowns
- ❖ CSS forms

Javascript:
- ❖ Including JavaScript in a Web page
- ❖ Using the Document Object Model (DOM) to access the HTML elements on the page dynamically
- ❖ Using JavaScript objects effectively, including the window object
- ❖ Creating and using variables

- ❖ Using event handlers to handle user-triggered events
- ❖ Using regular expressions with string methods to perform basic validation
- ❖ Working with primitive data types
- ❖ Passing arguments to a function
- ❖ Using conditional constructs and loops
- ❖ Debugging JavaScript code

JQuery:
- ❖ JQuery Introduction
- ❖ JQuery Basics
- ❖ JQuery Selectors
- ❖ Event Handling
- ❖ JQuery Effects
- ❖ JQuery AJAX
- ❖ Simple project to interact with JQuery + Ajax+ Servlet

AJAX:
- ❖ What is AJAX?
- ❖ Why do we required AJAX?
- ❖ How do we use AJAX?

Angular JS:
- ❖ Introduction to AngularJS
- ❖ Single Page Applications
- ❖ Controllers
- ❖ Models
- ❖ Views
- ❖ Expressions
- ❖ Filters
- ❖ Scopes
- ❖ Angular Forms
- ❖ Ajax, Data, and Angular
- ❖ Directives
- ❖ Simple AngularJS application + Servlets(or Spring)

Servlets
- ❖ Installation of Tomcat server.

- ❖ Web application directory structure.
- ❖ Introduction to static and dynamic web applications.
- ❖ Different types of input to the servlet. (request, config and context)
- ❖ Servlets architecture and life cycle.
- ❖ RequestDispatcher and sendRedirect
- ❖ Session Management (Session tracking)
- ❖ Filters
- ❖ Listeners.
- ❖ Wrappers
- ❖ Headers
- ❖ Declarative Security
- ❖ Database interaction (Various ways).

JSP

- ❖ Need of JSP,
- ❖ Scripting elements(expressions, scriptlets, declarations) and directives.
- ❖ JSP architecture and life cycle
- ❖ JSP actions (forward, include, param, useBean, setProperty, getProperty)
- ❖ JSP Custom tags Tag, IterationTag, BodyTag, TagSupport, Nested Tags, TagExtraInfo
- ❖ Expression Language (EL)
- ❖ JSTL
- ❖ Interacting with the database
- ❖ Web application design patterns (MVC, DAO, VO, BD, SL)

- ❖ Developing CRUD by using servlets, JSP and JDBC

Struts 2.0

- ❖ Introduction to Struts 2
- ❖ Web applications: a quick study
- ❖ Why use a framework for web applications
- ❖ The Struts2 framework and the declarative architecture
- ❖ The support of annotations
- ❖ **Actions and Interceptors**
  - ➢ The role of actions
  - ➢ Implementing and packaging actions
  - ➢ How to dialogue with data
  - ➢ The role of interceptors
  - ➢ Developing and declaring interceptors
  - ➢ The request/response life cycle
- ❖ **OGNL (Object-Graph Navigation Language)**
  - ➢ The common tasks in a web application
  - ➢ Data transfer and type conversion
  - ➢ The built-in converters
  - ➢ How to build a custom type converter
- ❖ **How to validate data**
  - ➢ The validation framework
  - ➢ Using the built-in rules
  - ➢ Writing a custom validator
  - ➢ Writing the actions for validation
- ❖ Internationalization
  - ➢ Java i18n and Struts2 framework
  - ➢ How to use Struts2 i18n
  - ➢ The default local determination

- ❖ <u>Struts2 integration and the persistence layer</u>
  Integrating Struts2 with Hibernate
- ❖ <u>Developing CRUD by using Struts2 and Hibernate</u>

Spring

- ❖ Introduction
- ❖ Spring IOC
- ❖ Spring AOP
- ❖ Spring DAO
- ❖ Spring ORM
- ❖ Spring MVC
- ❖ Spring Security
- ❖ Spring EL
- ❖ CRUD development by using Spring MVC and ORM.

Soap Based And REST FULL Webservice Tutorial

- ❖ Introduction
- ❖ Getting ready with software's
- ❖ SOAP UI Testing Tool
- ❖ Components of SOAP based Webservice

- ❖ Developing Webservice Provider
- ❖ Developing Webservice Consumer
- ❖ Components of SOAP based Webservice
- ❖ WSDL Explanation
- ❖ XML Explanation
- ❖ XSD Explanation
- ❖ SOAP
- ❖ JAXB API
- ❖ SDLC of web service
- ❖ REST FULL WEB SERVICE
  - ➢ Difference between SOAP AND REST
  - ➢ Getting start with RESTFULL WS
  - ➢ Developing REST FULL WS using JERSY FREMWORK
  - ➢ Developing rest full webservice provider
  - ➢ Developing rest full webservice client

List of the companies visited to Lara from past 4 months.

- ❖ **Professional Access**
- ❖ **Avekshaa**
- ❖ **SupaiInfotech**
- ❖ **Happiest Mind**
- ❖ **Surya Soft**
- ❖ **Analytics Quotient**
- ❖ **Revtech Solutions**
- ❖ **LendingKart Technologies**
- ❖ **Advent Systems**
- ❖ **Lean Infotech**

- ❖ **Key Falcon Solutions**
- ❖ **Dream Tekis Software Pvt. Ltd**
- ❖ **Amiti Software**
- ❖ **Azuga Software**
- ❖ **ModeFor Server**
- ❖ **Attocom**
- ❖ **Evolvus Solutions**
- ❖ **Mind Com Group**
- ❖ **Teminnova Technologies**
- ❖ **Boston Technology Corporations**

- Huawei Technologies India Pvt Ltd
- USR Infotech
- Extensio Software
- SYMPHONY TELECA
- CRM IT Solutions Pvt.Ltd.
- MicrohardInfotech
- Mist Minds
- Cloud Jet Labs
- Aqualogic Tech Systems Pvt. Ltd.
- Capiot
- People World
- SanchaInfotech
- Techno Soft
- Sapta Labs
- Ample Software
- QualNimbus
- Endure Software Solutions
- Greytip Technologies
- Globus Soft
- Therefore India
- BetterPlace Safety Solutions Pvt. Ltd.
- Estuate Software Private Ltd
- Vuram Technologies
- Algofusion Technologies India Pvt. Ltd.
- Riflexions Consulting Pvt.Ltd.
- IHS
- Replicon
- Tree Technologies
- Thinking Hut IT Solutions
- Selot Soft
- Prasanna Technologies
- Citus Tech
- CMS
- Brayons Soft Solutions (P).Ltd.
- IDC Technologies Solutons(I) Pvt. Ltd.
- Aspiring Wings Technologies Pvt. Ltd.
- Future Soft (India) Pvt. Ltd.
- Century Link
- Ibexi Solutions Pvt. Ltd.
- Manhattan Associates
- Atech Rays
- KNStek
- CGI
- IBell
- TurningPoint Software Solutions Pvt. Ltd.
- NTT Data
- Tetcos
- TransformEdge
- Software AG
- Fingertipplus
- Digital Harbor
- Vanward Technologies
- 3i Technology Solutions Pvt.Ltd
- Wifi Networks Pvt.Ltd.
- Zygnum
- Alchemy Solutions

- ❖ BSol Systems Pvt.Ltd.
- ❖ HireCraft Software Pvt. Ltd
- ❖ Trisys Software Technologies
- ❖ Diksha Technologies
- ❖ Izmo Ltd
- ❖ SPSS South Asia
- ❖ SrishtiESDM Pvt Ltd company
- ❖ Tataatsu IdeaLabs
- ❖ Startsavings
- ❖ BizRuntime
- ❖ Mastercom Technology
- ❖ Via.com
- ❖ Multifonds
- ❖ IOLITE TECHNOLOGIES
- ❖ ESCO