



# Lara Technologies

## A Unique future creator in Java



## Let us create world not Ad

**# 8, 2<sup>nd</sup> Floor, BTM Layout,  
2<sup>nd</sup> Stage, Bangalore - 560 076,  
Ph: 080 - 41680090, (+91) 9663646876,  
Website: [www.laratechnology.com](http://www.laratechnology.com)  
Mail id : [info@laratechnology.com](mailto:info@laratechnology.com)**

## Contents

Core Java .....	4
Core Java Part 1 .....	4
Core Java Part-2 .....	5
Exception Handling: .....	5
Assertions .....	6
Object Class.....	6
Multi Threading. ....	7
Strings .....	7
Arrays .....	9
Collection API: .....	9
List stream.....	9
Queue stream .....	10
Set stream .....	10
Map stream .....	10
Generics .....	11
File handling.....	11
JVM architecture .....	12
Design Patterns .....	12
Logical Coding, algorithms and Data Structures .....	13
Display Pattern.....	13
Number System.....	25
Arrays and Sorting Algorithms .....	27
Recursive Algorithm .....	29
String and File Handling.....	29
Data Structures .....	32
J2EE .....	33
SQL .....	33
JDBC .....	33

Hibernate .....	34
HTML .....	35
HTML5 .....	35
CSS .....	35
Javascript.....	36
JQuery .....	36
Servlets.....	36
JSP .....	36
Spring .....	37
Soap Based And REST FULL Web services Tutorial .....	37
(Angular4 Course Content).....	38
1. Getting Started.....	38
2. Type Script Fundamentals.....	38
3. Angular Fundamentals.....	38
4. Displaying Data and Handling Events.....	39
5. Building Reusable Components.....	39
6. Directives.....	39
7. Template - Driven Forms.....	40
8. Reactive Forms.....	40
9. Consuming HTTP Services.....	40
10. Routing and Navigation.....	41
List of the companies visited to Lara from past one year.....	42

# Core Java

## 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 built-in classes.,
6. println and print
7. Local Variable Introduction
8. unary operators
  - increment(pre & post)
  - decrement(pre & post)
9. Condition a 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
- Innerclasses
- 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
  - ArithmeticException
  - NumberFormatException
  - NullPointerException
  - ArrayIndexOutOfBoundsException
  - 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 RunnableInterface
- ❖ Thread with Threadclass
- ❖ 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()
- + operator
- concat()
- trim()
- charAt()
- indexOf()
- lastIndexOf()
- substring()
- split()
- toUpperCase()
- toLowerCase()
- equalsIgnoreCase()
- startsWith()
- endsWith()
- replace()
- replaceAll()
- "==" operator
- size()
- Differentiate equals and == operator etc..
- ❖ Explanation String memorymanagement
- ❖ StringBuffer class
  - Mutability
  - Capacity
  - Buffer
  - Thread safeness.
  - Extra methods like append(), reverse(), delete(), etc
- ❖ Differentiate String class and StringBuffer class
- ❖ StringBuilder class
- ❖ Differentiate StringBuffer andStringBuilder.
- ❖ Formatters
  - Flags, width, precision, conversion chars.
- ❖ Regular Expression
  - Patterns, Matcher
  - \d, \d+, \s, \w and so on
- ❖ StringTokenizer.
- ❖ Date, Calendar
- ❖ NumberFormat and DateFormatLocal.



## 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 forloop
  - Through Enhanced forloop
  - 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

### **JVM architecture**

- ❖ Class loader subsystem.
- ❖ Runtime data area
  - Method area
  - Heap
  - Java stacks
  - PC registers
  - Native method stacks
- ❖ Execution engine.

### **Design Patterns**

- ❖ Structural Design Patterns
- ❖ Behavioral Design Patterns

## Logical Coding, algorithms and Data Structures

### Display Pattern

1. \* \* \* \* \*  
\* \* \* \* \*  
\* \* \* \* \*  
\* \* \* \* \*  
\* \* \* \* \*

2. 1 1 1 1 1  
2 2 2 2 2  
3 3 3 3 3  
4 4 4 4 4  
5 5 5 5 5

3. 1 2 3 4 5  
1 2 3 4 5  
1 2 3 4 5  
1 2 3 4 5  
1 2 3 4 5

4. A A A A A  
B B B B B  
C C C C C  
D D D D D  
E E E E E

5. ABCDE  
ABCDE  
ABCDE  
ABCDE  
ABCDE

6. 5 5 5 5 5  
4 4 4 4 4  
3 3 3 3 3  
2 2 2 2 2  
1 1 1 1 1

7. 5 4 3 2 1  
5 4 3 2 1  
5 4 3 2 1  
5 4 3 2 1  
5 4 3 2 1

8. E E E E E  
D D D D D  
C C C C C  
B B B B B  
A A A A A

9. E D C B A  
E D C B A  
E D C B A  
E D C B A  
E D C B A

10. \*  
\* \*  
\* \* \*  
\* \* \* \*  
\* \* \* \* \*

11. 1  
2 2  
3 3 3  
4 4 4 4  
5 5 5 5 5

12. 1  
1 2  
1 2 3  
1 2 3 4  
1 2 3 4 5

13. A  
B B  
C C C  
D D D D  
E E E E E

14. A  
A B  
A B C  
A B C D  
A B D C E

15. \* \* \* \* \*  
\* \* \* \* \*  
\* \* \*  
\* \*  
\*

16. 1 1 1 1 1  
2 2 2 2  
3 3 3  
4 4  
5

20. 5 5 5 5 5  
4 4 4 4  
3 3 3  
2 2  
1

17. 1 2 3 4 5  
1 2 3 4  
1 2 3  
1 2  
1

21. 5 4 3 2 1  
5 4 3 2  
5 4 3  
5 4  
5

18. A A A A A  
B B B B  
C C C  
D D  
E

22. E E E E E  
D D D D  
C C C  
B B  
A

19. A B C D E  
A B C D  
A B C  
A B  
A

23. E D C B A  
E D C B  
E D C  
E D  
E

24.           \*  
              \* \*  
              \* \* \*  
              \* \* \* \*  
              \* \* \* \* \*

28.           A  
              AB  
              ABC  
              ABCD  
              ABCDE

25.           1  
              22  
              333  
              4444  
              55555

29.           \* \* \* \* \*  
              \* \* \* \* \*  
              \* \* \* \*  
              \* \* \*  
              \* \*  
              \*

26.           1  
              12  
              123  
              1234  
              12345

30.           55555  
              44444  
              33333  
              22222  
              11111

27.           A  
              BB  
              CCC  
              DDDD  
              EEEE

31.           12345  
              12344  
              12333  
              12222  
              12111  
              11111



32. E E E E E  
 D D D D  
 C C C  
 B B  
 A

36. 1  
 3 3 3  
 5 5 5 5 5  
 7 7 7 7 7 7  
 9 9 9 9 9 9 9 9

33. A B C D E  
 A B C D  
 A B C  
 A B  
 A

37. A  
 B B B  
 C C C C C  
 D D D D D D D  
 E E E E E E E E E

34. \*  
 \* \* \*  
 \* \* \* \* \*  
 \* \* \* \* \* \* \*  
 \* \* \* \* \* \* \* \* \*

38. A  
 C C C  
 E E E E E  
 G G G G G G G  
 I I I I I I I I I

35. 1  
 2 2 2  
 3 3 3 3 3  
 4 4 4 4 4 4 4  
 5 5 5 5 5 5 5 5 5

39. 1  
 1 2 3  
 1 2 3 4 5  
 1 2 3 4 5 6 7  
 1 2 3 4 5 6 7 8 9

40.           1  
              3 2 1  
              5 4 3 2 1  
              7 6 5 4 3 2 1  
              9 8 7 6 5 4 3 2 1

44.           A  
              B A B  
              C B A B C  
              D C B A B C D  
              E D C B A B C D E

41.           A  
              A B C  
              A B C D E  
              A B C D E F G  
              A B C D E F G H I

45.           1  
              1 2 1  
              1 2 3 2 1  
              1 2 3 4 3 2 1

42.           A  
              C B A  
              E D C B A  
              G F E D C B A  
              I H G F E D C B A

46.           A  
              A B A  
              A B C A B  
              A B C D A B C  
              A B C D E A B C D

43.           0  
              1 0 1  
              2 1 0 1 2  
              3 2 1 0 1 2 3  
              4 3 2 1 0 1 2 3 4

47.           \* \* \* \* \* \* \*  
              \* \* \* \* \*  
              \* \* \*  
              \*

48.           4 4 4 4 4 4 4  
              3 3 3 3 3  
              2 2 2  
              1

49. 7 7 7 7 7 7 7  
 5 5 5 5 5  
 3 3 3  
 1

50. 1 2 3 4 5 6 7  
 1 2 3 4 5  
 1 2 3  
 1

51. D D D D D D D D  
 C C C C C  
 B B B  
 A

52. G G G G G G G G  
 E E E E E  
 C C C  
 A

53. A B C D E F G  
 A B C D E  
 A B C  
 A

54. \*  
 \* \*  
 \* \* \*  
 \* \* \* \*  
 \* \* \*  
 \* \*  
 \*

55. 3  
 3 2  
 3 2 1  
 3 2 1 0  
 3 2 1  
 3 2  
 3

56. 3  
 2 3  
 1 2 3  
 0 1 2 3  
 1 2 3  
 2 3  
 1

57. D  
DC  
DCB  
DCBA  
DCB  
DC  
D

60. 3  
32  
321  
3210  
321  
32  
3

58. D  
CD  
BCD  
ABCD  
BCD  
CD  
D

61. 3  
23  
123  
0123  
123  
23  
3

59. \*  
\*\*  
\*\*\*  
\*\*\*\*  
\*\*\*  
\*\*  
\*\*  
\*

62. D  
CD  
BCD  
ABCD  
BCD  
CD  
D

63.           D  
               DC  
               DCB  
               DCBA  
               DCB  
               DC  
               D

64.           \*  
               \* \*  
               \* \* \*  
               \* \* \* \*  
               \* \* \* \* \*

65.           1  
               2 2  
               3 3 3  
               4 4 4 4  
               5 5 5 5 5

66.           1  
               1 2  
               1 2 3  
               1 2 3 4  
               1 2 3 4 5

67.           A  
               BB  
               CCC  
               DDDD  
               EEEE

68.           A  
               AB  
               ABC  
               ABCD  
               ABCDE

69.           \* \* \* \* \*  
               \* \* \* \*  
               \* \* \*  
               \* \*  
               \*

70.           5 5 5 5 5  
               4 4 4 4  
               3 3 3  
               2 2  
               1

71. 5 4 3 2 1  
 4 3 2 1  
 3 2 1  
 2 1  
 1

72. E E E E E  
 D D D D  
 C C C  
 B B  
 A

73. E D C B A  
 D C B A  
 C B A  
 B A  
 A

74. A B C D E  
 A B C D  
 A B C  
 A B  
 A

75. \*  
 \* \*  
 \* \* \*  
 \* \* \* \*  
 \* \* \* \* \*  
 \* \* \* \*  
 \* \* \*  
 \* \*
 \*

76. 1  
 2 2  
 3 3 3  
 4 4 4 4  
 5 5 5 5 5  
 4 4 4 4  
 3 3 3  
 2 2  
 1

77.        1  
          1 2  
          1 2 3  
          1 2 3 4  
          1 2 3 4 5  
          2 3 4 5  
          3 4 5  
          4 5  
          5

78.        1  
          1 2  
          1 2 3  
          1 2 3 4  
          1 2 3 4 5  
          1 2 3 4  
          1 2 3  
          1 2  
          1

79.        A  
          B B  
          C C C  
          D D D D  
          E E E E E  
          D D D D  
          C C C  
          B B  
          A

80.        A  
          A B  
          A B C  
          A B C D  
          A B C D E  
          B C D E  
          C D E  
          D E  
          E

81.       \*  
      \* \*  
      \* \*  
      \* \*  
      \* \*

85.           E  
          D D  
          C C  
          B B  
          A A

82.           1  
          2 2  
          3 3  
          4 4  
          5 5

86.       \*       \*  
          \*       \*  
          \*       \*  
          \* \*  
              \*

83.           5  
          4 4  
          3 3  
          2 2  
          1 1

87.       1       1  
          2       2  
          3       3  
          4 4  
          5

84.           A  
          B B  
          C C  
          D D  
          E E

88.       5       5  
          4       4  
          3 3  
          2 2  
          1



89.     A            A  
           B            B  
           C     C  
           D D  
           E

90.     E            E  
           D            D  
           C     C  
           B B  
           A

## Number System

1. Print bigger number from 2 given numbers.
2. Print smaller number from 3 given numbers.
3. Print middle number from 3 given numbers.
4. Print ascending order of 3 given numbers
5. Print true, if sum of any 2 numbers is a 3<sup>rd</sup> given number
6. Print true, if sum of squares of any 2 numbers is a 3<sup>rd</sup> given number
7. Find out given number is an even or odd?
8. Find out immediate next 5 multiple of a given number?
9. Print all odd numbers between two given numbers?
10. Find out sum of all even numbers between two given numbers?
11. Print java for multiples of 4 and print j2ee for multiples of 5 and if number is multiple of 4 and also 5, then print lara. Continue till to 100.
12. Find out whether digits are in raising order or not in a given number?
13. Swap two int variables without 3<sup>rd</sup> variable?
14. Find out total digits of a given number?
15. Find out sum of all digits of a given number?
16. Reverse a given number?
17. Find out given number is a palindrome or not?
18. Assume total number of digits in a given number is even. Reverse only first half of the given number
19. Assume given number containing 3 digits. now check it out first two digits sum is 3<sup>rd</sup> digit or not?
20. Find out factorial value for a given number?
21. Develop a program to find out given number is a prime or not?
22. Develop a program to print initial 20 prime numbers?
23. Develop a program to print prime numbers between 50 and 100?
24. Develop a program to print prime numbers between two given numbers?
25. Develop a program to print 20 prime numbers from 100.
26. Develop a program to print sum of initial 30 prime numbers?

27. Develop a program to print sum of prime numbers between two given numbers?
  28. Develop a program to print prime numbers in the reverse order from 70 to 20?
  29. Develop a program to print prime numbers which are immediately after multiples of 10 and below 200.
  30. Develop a program to print prime numbers which are just before multiples of 10 and below 200.
  31. Develop a program to print 30 prime numbers which are having digits in the raising order after 10. (Example: 13, 17, 19, 23, 29, 37, 39, 47, .....)
  32. Develop a program to print 30 prime numbers which are having digits in the descending order after 10. (Example: 31, 41, 43, 51, 53, 61, 71, 73, .....)
  33. Develop a program to print numbers which should come after 3 non prime numbers. Develop between 10 to 100.
  34. Print Fibonacci series till to 100.
  35. Print initial 10 numbers from the Fibonacci series.
  36. Print Fibonacci series from 100 to 10000?
  37. Print Fibonacci series in the reverse order from 5000 to 500?
  38. Print immediate next number in the Fibonacci series? Consider till to 5000
  39. Print all prime numbers only from the Fibonacci series? Continue till to 5000
  40. Find out given number is an Armstrong number or not?
  41. Find out initial 3 Armstrong numbers (Consider from 10)
  42. Find out all Armstrong numbers between 2 given numbers?
  43. Find out given number is a perfect number or not?
  44. Find out initial 5 perfect numbers (Consider from 10)
  45. Find out all perfect numbers between 2 given numbers?
  46. Find out given 2 numbers are anagrams or not  
Anagrams for 123 are 321, 312, 213, 231, 131
  47. Assume given number containing even number of digits. Reverse only 2<sup>nd</sup> half of the number?
  48. Find out sum of the digits of a given number?  
123 -> 6  
67 -> 13 -> 4  
869 -> 24 -> 6
  49. Move first half to second and second half to first? (total digits are even)  
Ex: 123456 convert to 456123,  
903512 convert to 512903
  50. Every digit swap with immediate digit. (total digits are even)  
Ex: 123456 convert to 214365
-

## Arrays and Sorting Algorithms

1. Find the sum of all given elements from an int array?
2. Find the min element from the given int array?
3. Find the max element from the given int array?
4. Find the 2<sup>nd</sup> min element from the given int array?
5. Find the 2<sup>nd</sup> max element from the given int array?
6. Find the average value of an int array?
7. Find out the sum of all even indexed elements from a given int array?
8. Find out the sum of all odd indexed elements from a given int array?
9. Find out the min value from all even indexed elements from a given int array?
10. Find out the max value from all odd indexed elements from a given int array?
11. Find out the avg value from all even indexed elements from a given int array?
12. Find out the avg value from all odd indexed elements from a given int array?
13. Find out the sum of all elements from a first half of given int array?
14. Find out the sum of all elements from a second half of given int array?
15. Find out the min value from a first half of given int array?
16. Find out the min value from a second half of given int array?
17. Find out the max value from a first half of given int array?
18. Find out the max value from a second half of given int array?
19. Find out the avg value from a first half of given int array?
20. Find out the avg value from a second half of given int array?
21. Read all elements from an array in the reverse order?
22. Read first half of the elements in the reverse direction from an array?
23. Read second half of the elements in the reverse direction from an array?
24. Read only even indexed elements from an array?
25. Read only even indexed elements from an array in the reverse order?
26. Read only odd indexed elements from an array?
27. Read only odd indexed elements from an array in the reverse order?
28. Find out an index of a specified element from a given array?
29. Swap two given indexed elements from the array?
30. Reverse the elements of given array?
31. Reverse only first half of the elements of given array?
32. Reverse only last half of the elements of given array?
33. Reverse only even indexed of the elements of given array?
34. Reverse only odd indexed of the elements of given array?
35. Swap odd indexed elements with its immediate next even indexed elements of given array?
36. Do right shift by one for elements of given array?

37. Do right shift by two for elements of given array?
38. Do right shift by three for elements of given array?
39. Do left shift by one for elements of given array?
40. Do left shift by two for elements of given array?
41. Do left shift by three for elements of given array?
42. Do right rotate by one for elements of given array?
43. Do right rotate by two for elements of given array?
44. Do right rotate by three for elements of given array?
45. Do left rotate by one for elements of given array?
46. Do left rotate by two for elements of given array?
47. Do left rotate by three for elements of given array?
48. Rotate first half of elements by one?
49. Rotate 2nd half of elements by one?
50. Rotate first half of elements by one and 2<sup>nd</sup> half of the elements by one separately?
51. Remove specified indexed element from the given array?
52. Update specified indexed element with a new element from the given array?
53. Remove all occurrences of specified element from the given array?
54. Remove range of elements from the given array
55. Remove all odd indexed elements from the given array?
56. Remove all even indexed elements from the given array?
57. Remove the duplicates from the given array?
58. Find out missed elements from the given array between min and max element
59. Remove elements of one array from another?
60. Retain one array elements in another array?
61. Find out common elements from two given arrays?
62. Find out uncommon elements from two given arrays?
63. Combine two arrays and develop a third array?
64. Find out index of an element which contains left indexed element is same as right indexed element.
65. Find out the elements which are not duplicates in the given array
66. Find out elements which are having minimum one duplicate?
67. Find out element frequency in the given array?
68. Combine two arrays and develop a third array. Consider element by element while combining
69. Combine two arrays and develop a third array. Consider element by element while combining and take forward direction from first array and reverse direction from 2<sup>nd</sup> array
70. Sorting int elements from an arrays? (use bubble sort)

71. Sorting int elements from an arrays? (use quick sort)
72. Sorting int elements from an arrays? (use merge sort)
73. Sorting int elements from an arrays? (use insertion sort)
74. Sorting int elements from an arrays? (use selection sort)

### Recursive Algorithm

1. Print sequential number from 1 to 100 without any loops
2. Print sequential number from 100 to 1 without any loops
3. Print prime numbers from 50 to 100 without any loops
4. Develop Fibonacci series without loops
5. Reverse a string without any loops
6. Find out sum of all the digits in a given number without loops
7. Reverse a number without any loops?
8. Find out factorial value for a given number without loops?

### String and File Handling

1. Write a program to reverse a String
2. Write a program to reverse a String with recursive algorithm?
3. Write a program to reverse first half separately and 2<sup>nd</sup> half separately?
4. Write a program to rotate one char in a given string
5. Find out length of the string without length() method of a String?
6. **Find out how many words are there in a given string?**
7. **Write a java program to find the duplicate words and their number of occurrences in a string?**
8. **Write a program to reverse the given string word wise?**
9. **Rotate the string word wise by one**
10. **Write a java program to count the total number of occurrences of a given character in a string?**
11. **Write a java program to count the number of occurrences of each character in a string?**
12. **Write a java program to remove all white spaces from a string?**
13. **Write a program to check whether given string is a palindrome or not?**
14. **Write a program to check whether given two strings are anagrams?**
15. **Write a program to check the balance of brackets in the given string?**
16. **Write a java program to find duplicate characters in a string?**
17. **Write a java program to check whether one string is a rotation of another?**
18. **Write a java program to reverse a given string with preserving the position of spaces?**
19. **Write a java program to reverse each word of a given string?**

20. Write a java program to find the percentage of uppercase letters, lowercase letters, digits and special characters in a given string?
21. How do you find longest substring without repeating characters in the given string?
22. How do you swap two string variables without using third or temp variable in java?
23. Write a java program to find all permutations of a string?
24. How do you find first repeated and non-repeated character in the given string in java?
25. How do you find the number of characters, words and lines in the given text file in java?
26. How do you find the most repeated word in a text file in java?
27. How to search a word inside a string?
28. How to remove html tags from a string?
29. Write a program to print all permutations of String?
30. Write a function to find out longest palindrome in a given string?
31. Write a program to validate email format?
32. Write a program to validate date format?
33. Write a program to validate phone number format?
34. Write a program to validate specified username format?
35. Write a program to validate specified password format?
36. Write a program to validate hex code format?
37. Write a program to validate image file extension?
38. Write a program to validate IP Address
39. Write a program to validate 12 hours specified time format?
40. Write a program to validate 24 hours specified time format
41. Find out longest string from the given file?
42. Find out longest sentence from the given file?
43. Find out number of lines in the given file?
44. Find out number of words in a given file?
45. Find out a word which is occurring more times in a file?
46. Find out a word count of all the words from a given files?
47. Find out total number of files and sub directories from a given directory?
48. Given a string, look for a mirror image (backwards) string at both the beginning and end of the given string. In other words, zero or more characters at the very beginning of the given string, and at the very end of the string in reverse order (possibly overlapping). For example, the string "abXYZba" has the mirror end "ab".

mirrorEnds("abXYZba") → "ab"

mirrorEnds("abca") → "a"

mirrorEnds("aba") → "aba"

49. Given a string, return the sum of the numbers appearing in the string, ignoring all other characters. A number is a series of 1 or more digit chars in a row. (Note: Character.isDigit(char) tests if a char is one of the chars '0', '1', .. '9'. Integer.parseInt(string) converts a string to an int.)
- ```
sumNumbers("abc123xyz") → 123  
sumNumbers("aa11b33") → 44  
sumNumbers("7 11") → 18
```
50. Given a string, return the sum of the digits 0-9 that appear in the string, ignoring all other characters. Return 0 if there are no digits in the string. (Note: Character.isDigit(char) tests if a char is one of the chars '0', '1', .. '9'. Integer.parseInt(string) converts a string to an int.)
- ```
sumDigits("aa1bc2d3") → 6  
sumDigits("aa11b33") → 8  
sumDigits("Chocolate") → 0
```
51. Given a string, compute a new string by moving the first char to come after the next two chars, so "abc" yields "bca". Repeat this process for each subsequent group of 3 chars, so "abcdef" yields "bcaefd". Ignore any group of fewer than 3 chars at the end.
- ```
oneTwo("abc") → "bca"  
oneTwo("tca") → "cat"  
oneTwo("tcagdo") → "catdog"
```
52. Given a string, return a string where for every char in the original, there are two chars.
- ```
doubleChar("The") → "TThhee"  
doubleChar("AAAb") → "AAAAbbbb"  
doubleChar("Hi-There") → "HHii--TTtheerree"
```
53. Given two strings, **a** and **b**, create a bigger string made of the first char of a, the first char of b, the second char of a, the second char of b, and so on. Any leftover chars go at the end of the result.
- ```
mixString("abc", "xyz") → "axbycz"  
mixString("Hi", "There") → "HTihere"  
mixString("xxxx", "There") → "xTxhxexre"
```
54. Given a string and an int n, return a string made of n repetitions of the last n characters of the string. You may assume that n is between 0 and the length of the string, inclusive.
- ```
repeatEnd("Hello", 3) → "llollollo"  
repeatEnd("Hello", 2) → "lolo"  
repeatEnd("Hello", 1) → "o"
```



55. Return a version of the given string, where for every star (\*) in the string the star and the chars immediately to its left and right are gone. So "ab\*cd" yields "ad" and "ab\*\*cd" also yields "ad".

starOut("ab\*cd") → "ad"

starOut("ab\*\*cd") → "ad"

starOut("sm\*eilly") → "silly"

56. Given a string and a non-empty **word** string, return a version of the original String where all chars have been replaced by pluses ("+"), except for appearances of the word string which are preserved unchanged.

plusOut("12xy34", "xy") → "++xy++"

plusOut("12xy34", "1") → "1++++"

plusOut("12xy34xyabcxy", "xy") → "++xy++xy+++xy"

## Data Structures

1. LinkedList : Adding an Element
2. LinkedList : Adding array of Elements
3. LinkedList : Insertion of an element in the specified position
4. LinkedList : Insertion of an array of elements in the specified position
5. LinkedList : Iterating all elements
6. LinkedList : updating a specified Element with a new element.
7. LinkedList : Finding out length (size)
8. LinkedList : Removing specified element
9. LinkedList : Removing specified position element
10. LinkedList : Removing all duplicate data nodes.
11. LinkedList : swapping two specified node's data
12. LinkedList : swapping two specified nodes itself
13. LinkedList : Reverse data from the nodes
14. LinkedList : Reverse all nodes itself
15. LinkedList : Sort Data from the nodes
16. LinkedList : Sort Nodes itself
17. Merge two sorted linkedlists
18. Merge Sort for LinkedLists
19. Detect and Remove Loop in a Linked List
20. Rotate a LinkedList
21. Making LinkedList as a circular
22. Split a Circular LinkedList into two halves
23. Developing DoublyLinkedList
24. Delete a node in a Doubly LinkedList
25. Reverse a Doubly LinkedList
26. Making Doubly LinkedList as a circular
27. BinaryTree : Add elements
28. BinaryTree : Reading InOrder
29. BinaryTree : Reading PostOrder
30. BinaryTree : Reading PreOrder
31. BinaryTree : Search
32. BinaryTree : Count Nodes



## J2EE

### 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 relationship
  - 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
  - ResultSet
  - ResultSetMetadata
  - DataBase MetaData
  - 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
  - ResultSets scrollability
  - ResultSet Updatability.
  - Batch Updates

- Storing and reading Image/audio/video file from the database
- ❖ JDBC 3.0 features
  - Retrieval of auto generated keys.
  - ParameterMetaData.
  - SavePoint
- ❖ JDBC 4.0 features
  - ROWID datatype
  - 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
  - 1<sup>st</sup> level and 2<sup>nd</sup> 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 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

## 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

- ❖ 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

- ❖ How do we use AJAX?

### 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

### 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).

### 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 require AJAX?

### 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
- ❖ **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

## Spring

- ❖ Introduction
- ❖ Spring IOC
- ❖ Spring AOP
- ❖ Spring DAO
- ❖ Spring ORM
- ❖ Spring MVC
- ❖ Spring Security

- ❖ Spring EL
- ❖ CRUD development by using Spring MVC and ORM.

## SoapBasedAndRESTFULL Webservice Tutorial

- ❖ Introduction
- ❖ Getting ready with software's
- ❖ SOAP UI Testing Tool
- ❖ Components of SOAP based Webservices
- ❖ Developing Webservices 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 FREAMWORK
  - Developing restfull webservice provider
  - Developing restfull webservice client

## Angular4 Course Content

### ❖ Getting Started

- Angular Version History
- What is Angular
- Setting Up the Development Environment
- Architecture of Angular Apps
- Your First Angular App
- Structure of Angular Projects

### ❖ Type Script Fundamentals

- Introduction
- What is TypeScript
- Your First TypeScript Program
- Declaring Variables
- Type
- Type Assertions
- Arrow Functions
- Interface
- Classes
- Objects
- Constructors
- Access Modifiers
- Access Modifiers in constructor parameters
- Properties
- Modules

### ❖ Angular Fundamentals

- Introduction
- Building Blocks of Angular Apps
- Components
- Dependency Injection
- Generating Components Using Angular CLI
- Templates
- Directives
- Services
- Generating Services Using Angular CLI

## ❖ Displaying Data and Handling Events

- Introduction
- Property Binding
- Attribute Binding
- Adding Bootstrap
- Class Binding
- Style Binding
- Event Binding
- Event Filtering
- Template Variables
- Two-way Binding
- Pipes
- Custom Pipes

## ❖ Building Reusable Components

- Introduction
- Component API
- Input Properties
- Aliasing Input Properties
- Output Properties
- Passing Event Data
- Aliasing Output Properties
- Templates
- Styles
- View Encapsulation
- ng- content
- ng-container

## ❖ Directives

- Introduction
- ngIf
- Hidden Property
- The Leading Asterisk
- ngClass
- ngStyle
- ngSwitchCase
- ngFor
- ngFor and Change Detection
- ngFor and trackBy
- Safe Traversal Operator
- Creating Custom Directives

## ❖ Template-Driven Forms

- Introduction
- Building a Bootstrap Form
- Types of Forms
- ngModel
- Adding Validation
- Specific Validation Errors
- Styling Invalid Input Fields
- Cleaner Templates
- ngForm
- ngModelGroup
- Control Classes and Directives
- Disabling the Submit Button
- Working with Check Boxes
- Working with Drop-down Lists
- Working with Radio Buttons

## ❖ Reactive Forms

- Introduction
- Building a Bootstrap Form
- Creating Controls Programmatically
- Adding Validation
- Specific Validation Error
- Implementing Custom Validation
- Asynchronous Operations
- Asynchronous Validation
- Showing a Loader Image
- Validating the Form Upon Submit
- Nested FormGroup
- FormArray
- FormBuilder

## ❖ Consuming HTTP Services

- Introduction
- JSONPlaceholder
- Getting Data
- Creating Data
- Updating Data
- Deleting Data
- OnInit Interface
- Separation of Concerns
- Extracting a Services
- Handling Errors
- Handling Unexpected Errors



- Handling expected Errors
- Throwing Application-specific Errors
- Handling Bad Request Errors
- Importing Observables operators and Factory methods
- Global Error Handling
- Extracting a Reusable Error Handling method
- Extracting a Reusable Data Service
- The Map Operator
- Optimistic vs Pessimistic Updates
- Observables vs Promises

#### ❖ Routing and Navigation

- Introduction
- Routing in a Nutshell
- Configuring Routes
- RouterOutlet
- RouterLink
- RouterLinkActive
- Getting the Route Parameters
- Why root Parameters are
- Observable
- Routes with Multiple Parameters
- Query Parameters
- Subscribing to Multiple Observables
- Switch Map Operator
- Programmatic Navigation

## List of the companies visited to Lara from past one year.

- ❖ Professional Access
- ❖ Avekshaa
- ❖ Supainfotech
- ❖ 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
- ❖ Sanchalinfotech
- ❖ 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.
- ❖ Reflexions 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**