**LAKSHMAN\_JAVA\_IT\_SCHOOL**



CORE JAVA = JSE, J2SE

## What is Language?

1. **What is programming Language?**

## Types of Programming languages’?

* 1. Machine Language
  2. Assembly Language
  3. High Language

## Difference Product and Project.

1. **Procedure Oriented Programming Language.**

## What is Object Oriented Programming Language.

1. **Introduction to Java**

## Principles of Java

* 1. Simple
  2. Platform Independency
  3. Architectural Neutral
  4. Portable
  5. High performance
  6. Robust
  7. Secure
  8. Multithreaded
  9. Distributed
  10. Dynamic
  11. Oops

## Oops Introduction

* 1. Abstraction
  2. Encapsulation
  3. Inheritance
  4. Polymorphism

## Java Programming Elements

* 1. Identifiers and Naming conventions
  2. Keywords

## Operators

* + - Assignment operator
    - Arithmetic operators
    - Conditional operator
    - Increment/Decrement operators
    - Relational operators
    - Logical operators
    - Bitwise operators
    - Shift operators
* Compound assignment operators
  1. Separators

### Literals

1. Integral Literals
2. Floating Literals
3. Character Literals
4. Boolean Literals
5. String Literals
6. Underscore in integral literals

### Comments

* Single level Comment
* Multi level Comment
* Document Comment

### Data types

1. **Primitive data types**
   1. Byte, short, int, long, float, double, char, Boolean

### Referenced Variables

* 1. Arrays
  2. Class
  3. Interface
  4. Enum
  5. Annotation
  6. Escape Characters

### Casting

* 1. Implicit Casting
  2. Explicit Casting

### Arrays

* 1. Single Dimensional arrays
  2. Double Dimensional arrays
  3. Multi Dimensional arrays
  4. Anonymous arrays
  5. Jagged arrays

### Statements

* 1. Sequential statements

### Control Statements

1. **Conditional control statements**

### Loop control statements

* + for
  + while
  + do..while
  + Enhanced for loop/for-each loop

### Branching statements

**Switch with String parameter**

### Variables and type of variables

1. **Class level Variables**
   1. Static variables
   2. Non-static variables

### Local Variables

1. **Blocks**
   1. Static blocks
   2. Non-static blocks

### Methods and type of methods,

* 1. Static methods
  2. Non-static methods
  3. Void methods
  4. Non-void methods
  5. Parameterized methods
  6. Non-parameterized methods
  7. Abstract Methods
  8. Concrete Methods
  9. Variable arguments in method parameters
  10. Default methods in interface
  11. Static methods in interface

### Wrapper classes

* 1. Number
  2. Byte
  3. Short
  4. Integer
  5. Long
  6. Float
  7. Double
  8. Character
  9. Boolean

### Conversions

* 1. Auto Boxing
  2. Auto Un Boxing

### Packages

* 1. How to create single package
  2. How to sub packages
  3. How to package in other directory
  4. How to Access other packages
  5. Static import statement
  6. General import statement
  7. Difference between #include and import statements

### Accessibility Modifiers

* 1. Private
  2. Protected
  3. (package private) or default
  4. public

### Jar and its handling

1. **How to create batch files**

### Data Hiding

1. **Data Abstraction**

### Encapsulation

1. **Command Line Arguments**

### Java.util.Scanner class and its method

1. **Inheritance and Types of Inheritance**
   1. Single Level Inheritance
   2. MultiLevel Inheritance
   3. Hierarchal Inheritance
   4. Multiple Inheritance
   5. Hybrind Inheritance

### Interfaces

* General interface
* Functional interface
* Marker/tagling interface

### Abstract Classes

1. **Polymorphism**
   1. Static polymorphism
   2. Dynamic polymorphism
   3. Method overloading
   4. Method Overriding
   5. Covariant return types
   6. Method Hiding

### Inner Classes

* 1. Non-static inner class/simple inner class
  2. Static inner classes
  3. Method level inner classes
  4. Anonymous inner classes

### Java.lang.Object

* 1. getClass()
  2. toString()
  3. hashCode()
  4. equals

### clone()

1. Deep cloning
2. Shallow cloning

### JVM architecture

1. **String Handling**
   1. Introduction to Strings
   2. Creating objects to String
   3. String Constant Pool
   4. String library functions
   5. Mutable objects
   6. Immutable objects
   7. String/StringBuffer/StringReader
   8. Creating Immutable class

### Exception Handling

* 1. Introduction to Error and Exception and Syntax Errors

### Types of Exceptions

1. Checked exceptions
   * Fully Checked Exceptions
   * Partially Checked Exceptions
2. Un checked exceptions
   1. Try, catch, throw, throws, finally
   2. Nested try blocks
   3. Multiple catch blocks
   4. Handling exceptions
   5. User defined exceptions
   6. Chained Exceptions
   7. Try with resource
   8. Catch block with multiple exceptions.

### IOStreams

* 1. What is Stream

### Types of Streams

1. **Byte Streams**

### Character Streams

* 1. FileOutputStream
  2. FileInputStream
  3. DataOutputStream
  4. DataInputStream
  5. FileWriter
  6. FileReader
  7. InputStreamReader
  8. Serialization
  9. De serialization
  10. Customization
  11. Externalization
  12. PrintStream
  13. System.out.println
  14. Console

### Multithreading

* 1. Introduction to multi tasking and multi threading
  2. Drawbacks in multi tasking
  3. Creation of Thread
  4. Life cycle of Thread
  5. Thread class
  6. Runnable interface
  7. Constructors of Thread class.
  8. Inline Thread Creation
  9. Priorities of threads.
  10. Naming to threads.
  11. Synchronization
  12. sleep(),join(), wait(), notify(), notifyAll(),
  13. TheadGroup
  14. DeadLock
  15. ThreadPoll introduction
  16. ExecutorFrameWork
  17. ThreadLocal
  18. RseentrantLock

### Net Working

* 1. Introduction to networks
  2. Types of networks
  3. Client
  4. Server
  5. Client machine
  6. Server machine
  7. Request
  8. Response
  9. IP Address
  10. Port
  11. Socket
  12. Client –server architecture
  13. Socket programming example

### Collection Framework and Generics

* 1. Introduction to collections
  2. Introduction to generics
  3. Difference between arrays and Collections
  4. Collection interfaces

### List Interface

* + ArrayList
  + LinkedList
  + Vector
  + Stack

### Set Interface

* + Hashtable
  + HashSet
  + LinkedHashSet
  + SortedSet
  + NavigableSet
  + TreeSet

### Map Interface

* + Dictionary
  + HashTable
  + Properties
  + HashMap
  + LinkedHashMap
  + IdentityHashMap
  + WeakHashMap
  + SortedMap
  + NavigableMap
  + TreeMap

### Queue Interface

1. LinkedList
2. PriorityQueue

### BlockingQueue

* + LinkedBlockingQueue
  + PriorityBlockingQueue
  1. Collections Class
  2. Arrays Class
  3. Enumerations
  4. Iterator
  5. ListIterator
  6. Comparator
  7. Comparable
  8. Java.util.Stream api
  9. New date and time api
  10. Java.util.concurent package introductions and related implementation classes information

1. CopyOnWriteArrayList
2. CompyOnWriteArraySet
3. ConcurrentHashMap

### Date and Formatting text, Random, StringTokenizer

1. **Internationalization.**

### API documentation and How to use it.

1. **Annotations**

### Meta annotations

* + - Target
    - Retention
    - Inherited
    - Documented
    - Repeatable
    - Native

### Standard annotations

* + - Deprecated
    - Override
    - SuppressWarnings
    - SafeVarargs
    - FunctionalInterface

### Reflections API

Java.lang.Class Java.lang.reflect.Method Java.lang. reflect.Field Java.lang. reflect.Modifier

Java.lang. reflect.Constructor

### RegularExpression

* 1. Pattern
  2. Matcher

1. **Enums**

# Java 8 features:

1. Lambda Expressions.
2. Method References.
3. Functional interface.
4. Predefine functional interface
5. Stream API.
6. Default Methods in interface.
7. Static methods in interface
8. Collectors class.
9. Optional Class.
10. LocalDate,LocalTime,LocalDateTime,Perio d,Year
11. Type inference
12. StringJoiner class.
13. Nashron javascript engine
14. Base64 encode and decode.
15. Parllel array sort.
16. Parameter reflection.

# Java 9 features:

1. Jigsaw
2. JShell
3. Try with resource extra configuration
4. Factory methods in collection
5. Private methods in interface
6. Enhancements in stream api
7. Http2 client
8. G1 Garbage collector