

# **Core Java with Complete logical programming**

**Srinivas G**

**Duration : 40 days**

1. Introduction to computer languages?
2. Need of computer languages?
3. Java V/S other languages
4. Platform dependency?
5. Introduction to Java?
6. History of Java
7. JDK includes
8. Versions and Certifications
9. Platform Independency?
10. Structure of Java-application?
11. Contexts of Java application
  
12. Introduction to Variable, Blocks and Methods ?
  - Introduction to variables and identifiers
  - Type of variables
  - Memory construction and destruction of variables
  - Rules to create variables
  - Naming conventions of variables
  - Blocks introduction
  - Methods introduction.
  
- 11) Datatypes
  - Primitive types
    - Introduction
    - Memory occupancy
    - Ranges
    - Errors(compile time and runtime)
    - Primitive type casting(implicit and explicit)
  - Reference types
  
13. First java application?
  - Class keyword and its naming convention
  - Class naming convention
  - Why main has public access
  - Why main has static modifier
  - Why return type is void
  - System.out.println() description
  - Naming convention of methods and packages.
  - Overloading main method
  
14. Operators
  - Assignment operator

- Arithmetic operators
- Conditional operator
- Increment/Decrement operators
- Relational operators
- Logical operators
- Bitwise operators
- Shift operators
- Compound assignment operators

#### 15. Control Statements

- Conditional control statements
  - If
  - If-else
  - If-else-if
  - Nested if
  - switch
- Loop control statements
  - While
  - Do-while
  - For loop
  - Nested loops
- Branching statements
  - break;
  - continue
  - return
  - exit

#### 16. Methods

- Need of functions.
- Types of functions
  - Pre-defined
  - User-defined
- Classification of functions
  - No args-no return values function
  - With args-no return values function
  - With args-with return values function
  - No args with return values function
  - Recursive function.

#### 17. Class members in java

- a. Static members(class level)
  - i. Static block
  - ii. Static variables
  - iii. Static methods
  - iv. Main method
  - v. Memory allocation using JVM architecture
  - vi. Accessing Static members in multiple classes
  - vii. Class loader usage

- b. Non-static members(object level)
  - i. Non-static block
  - ii. Non-static variables
  - iii. Non-static methods
  - iv. Constructor
  - v. Introduction to “this” keyword.
  - vi. Memory allocation using JVM architecture
  - vii. Accessing Non-static members from multiple classes

#### 18. Wrapper classes

- Introduction
- Why wrapping
- Boxing and Unboxing
- Auto Boxing and Auto Unboxing
- Primitive to String conversion (using valueOf()and toString() methods)
- String to Primitive conversion (using valueOf() and xxxValue() methods)

#### 19. Packages

- Introduction
- How to create user defined package
- Calling members of same package
- Calling members of another package
- Need of import statement
- Need of fully qualified name
- Difference between import and include
- Creating Sub packages
- Importing sub packages
- Access specifiers introduction
  - public
  - private
  - protected
  - <package> or <default>
- Usage of access specifiers in packages

#### 20. Command line arguments

- Introduction
- Advantage and disadvantages
- parseXxx methods
- Scanner class
- Random class

#### 21. OOPS

- a. Introduction to OOPS
- b. Introduction to class, object
- c. Encapsulation, Abstraction
- d. Inheritance
  - i. ‘this’ keyword

- Initialization of object using “this”
  - Accessing variable using “this”
  - Invoking constructor using “this()”
  - Constructor chaining
  - “this” as parameter to a function
  - Returning “this” to a function
  - Calling instance members from non-static context
- ii. ‘super’ keyword
- Initialization of super class object
  - Constructor chaining in hierarchy
  - Instantiation of abstract class
  - Accessing super class variable using “super”
  - Invoking super class constructor using “super()”
- iii. Types of inheritance
- Single(simple) inheritance
  - Multi level inheritance
  - Hierarchical inheritance
  - Applying Modifiers and Access specifiers in inheritance
- iv. Object casting
- Implicit & Explicit up casting
  - Implicit & Explicit down casting
  - Data hiding (static and non-static)
  - Introduction to dynamic polymorphism
- e. final keyword
- ◆ if class is final
  - ◆ if method is final
  - ◆ if variable is final
  - ◆ if abstract class is final
  - ◆ if interface is final
- f. Polymorphism
- ◆ Static polymorphism(method over loading)
  - ◆ Runtime polymorphism(method overriding)
- g. Abstract classes
- Introduction to Abstraction
  - Use of Abstraction
  - Instantiation of abstract class
  - Illegal combination of modifiers
  - Initialization of abstract class object
  - Super-Sub construction chaining
  - Runtime polymorphism in abstraction
- h. Interfaces
- Introduction to interfaces
  - Use of interfaces
  - Extending interfaces
  - Implementing interfaces
  - Multiple inheritance in java

- i. Aggregation
- j. Association
- k. Composition
- l. Singleton class
- m. Factory class.

## 22.Exception handling

- Introduction
- Why exceptions
- Exceptions API
- Checked exceptions
- Un checked exceptions
- Try, catch, throw, throws, finally
- Nested try blocks
- Multiple catch blocks
- Cautions while handling exceptions
- Handling checked and unchecked exceptions
- Throwing pre-defined exception class object explicitly
- User defined exceptions
- Throwing user-defined exception objects.
- Finally block usage.

## 23.Multi threading

- Introduction to multi tasking and multi threading
- Drawbacks in multi tasking
- Creation of Thread
- Life cycle of Thread
- Threads Using Thread class
- Threads Using Runnable interfaces
- Constructors of Thread class.
- Time management in multi tasking and multi threading
- Priorities of threads.
- Naming to threads via constructors or via setters.
- Synchronization
- sleep(),join(), wait(), notify(), notifyAll(),

## 24.Garbage collection

- Introduction GC
- Introduction to daemon threads
- Drawbacks of
- finalize()
- System.gc()
- Runtime.gc()
- exec();
- JVM memory increment
- Process class.

## 25.io streams

- Byte streams
- Character streams
- Object streams (Serialization)
- Buffered streams
- Working with files
  - ◆ Creating a file
  - ◆ Creating a directory
  - ◆ Creating path
  - ◆ Deleting file or directory
  - ◆ Hiding file
  - ◆ Setting read, write and execute permissions to files

## 26. Reflection API

## 27. Inner classes

- Static inner classes
- Non-static inner classes
- Local inner classes
- Anonymous inner classes

## 28. Strings

- Introduction to Strings
- Creating objects to String
- String library functions
- Mutable objects
- Immutable objects
- String/StringBuffer/StringReader
- Creating Immutable class

## 29. Collections

- Introduction to collections
- Introduction to generics
- Difference between arrays and Collections
- Collection interfaces
- List Interface
  - ArrayList
  - Vector
  - LinkedList
- Set Interface
  - HashSet
  - Introduction to Hashtable
  - Load factor influence
  - LinkedHashSet
  - TreeSet
- Map Interface
  - HashMap
  - LinkedHashMap
  - TreeMap
- Queue Interface
  - LinkedList

- PriorityQueue
- Time management over collection classes
- Memory management over collection classes
- Deleting duplicate objects from collections
- equals() method
- Iterator class
- Enumeration class
- for-each loop
- influence of generics in collections

### 30.JDK new features

### 31.AWT, Applets and Swings

- Introduction to AWT
- AWT library classes and interfaces
- Introduction to Applets
- Creating Applets
- Applet Life cycle
- Drawing shapes and colors in Applets
- Creating components
- LayoutManagement
  - Flow layout
  - Grid layout
  - Box layout
  - Border layout
- Event Listeners
  - ActionListener
  - TextListeners
  - MouseListeners
  - MouseMotionListeners
  - FocusListeners
- Calculator program
- Introduction to swings
- Swings API
- Light weight components(independent)
- Heavy weight components(native or dependant)
- Components creation
- Menu creation
- Sub menu creation
- Setting Listeners to components