

# Core Java Course Content

## Course Content:

1. Introduction to computer languages?
2. Need of computer languages?
3. Java V/S other languages
4. Platform dependency and Platform Independency?
5. Introduction to Java?
6. JDK includes
7. Structure of Java-application?
8. 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
9. Data types in java?
  - Introduction to variables and identifiers
  - Type of variables
  - Memory construction and destruction of variables
  - Rules to create variables
  - Naming conventions of variables
  - Primitive types
    - Introduction
    - Memory occupancy
    - Ranges
    - Errors(compile time and runtime)
    - Primitive type casting(implicit and explicit)
  - Reference types
10. Operators
  - Assignment operator
  - Arithmetic operators
  - Conditional operator
  - Increment/Decrement operators

- Relational operators
  - Logical operators
  - Bitwise operators
  - Shift operators
  - Compound assignment operators
11. Control Statements
- Conditional control statements
  - Loop control statements
  - Branching statements
12. Functions
- 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.
13. 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
14. 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)

#### 15. 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

#### 16. Command line arguments

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

#### 17. For-each loop

#### 18. 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. Polymorphism
  - ◆ Static polymorphism(method over loading)
  - ◆ Runtime polymorphism(method overriding)
- f. 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
- g. Interfaces
  - Introduction to interfaces
  - Use of interfaces
  - Extending interfaces
  - Implementing interfaces
  - Multiple inheritance in java

- h. Aggregation
  - i. Association
  - j. Composition
  - k. Singleton class
  - l. Factory class.
19. 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.
20. 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(),
21. Serialization
- Why serialization
  - Transient keyword
  - What is persistent state of an object
  - Marker interface

- Can we serialize subclass object if super class is not Serializable
  - Can we serialize an object contains reference to non-serializable object
22. Object cloning
23. Garbage collection
- Introduction GC
  - Introduction to daemon threads
  - Drawbacks of
  - finalize()
  - System.gc()
  - Runtime.gc()
  - exec();
  - JVM memory increment
  - Process class.
24. Io streams
- Byte streams
  - Character streams
  - Object streams
  - Byte array streams
  - Data input streams
  - 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
25. Reflection API
26. Inner classes
- Static inner classes
  - Non-static inner classes
  - Local inner classes
  - Anonymous inner classes
27. Strings
- Introduction to Strings
  - Creating objects to String
  - String library functions
  - Mutable objects

- Immutable objects
- String/StringBuffer/StringReader
- Creating Immutable class

## 28. 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
  - LinkedHashMap
  - 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