**Core Java**

**Core Java Part 1**

1. JDK

Downloading, installation and hello world development.

2. PATH environmentVariable

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 globalvariables

13.Introduction ofSIB

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 keyword.

35.CLASSPATH

36.javap, javadoc, jar commands.

37.Eclipse

38.Enhanced forloop

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 finallyblock.

❖ Return statement insidetry/catch/finally

❖ Errors andExceptions

❖ Checked and Unchecked

❖ Explanation of Error types

➢ StackOverFlowError

➢ OutOfMemoryError

➢ NoClassDefinationFoundError

➢ NoSuchMethodError etc…

❖ Explanation of UncheckedException types

➢ ArithmaticException

➢ NumberFormatException

➢ NullPointerException

➢ ArrayIndexOutOfBoundException

➢ ClassCastException

❖ 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 catchblocks

❖ throw keyword and its usage

❖ Developing user defined exception class.

❖ Differentiate throws andthrow.

**Assertions**

❖ Why do we require assertions.

❖ enablinganddisablingthe assert statements inexecutionwise,package

wise and classwise.

❖ Types of assertstatements.

❖ 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 ofthreads.

❖ 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 basicinformation.

❖ 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 StringBufferclass

❖ 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 dimensionalArray

❖ Multi dimensionalArray

❖ Java.util.Arrays

❖ Comparable Interface

❖ Comparator Interface

**Collection API:**

❖ Limitations of arrays.

❖ Introduction to CollectionAPI

❖ 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 ownThreadLocal

❖ Developing Object Pool Design pattern

❖ Synchronization, developing our own synchronized collections.

❖ Fail fast and Fail Safe

❖ Concurrent package

**Generics:**

➢ Why UseGenerics?

➢ Generic Types

➢ Raw Types

➢ Generic Methods

➢ Bounded TypeParameters

➢ Generic Methods and Bounded Type Parameters

➢ Generics, Inheritance,and Subtypes

➢ Type Inference

➢ Wildcards

➢ Upper BoundedWildcards

➢ Unbounded Wildcards

➢ Lower BoundedWildcards

➢ Wildcards andSubtyping

➢ Wildcard Capture and Helper Methods

➢ Guidelines for WildcardUse

➢ Type Erasure

➢ Erasure of GenericTypes

➢ Erasure of GenericMethods

➢ 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 DesignPatterns

❖ Behavioral DesignPatterns