NARESH i technologies

Course Content Core Java [85 sessions]

Index	Topics
1	Introduction to Language
2	Introduction to Java
	a) Flavors of Java, History of Java, Features of Java, Introduction toJDK, JRE, JVM and JIT Compiler.
3	Platform Independency in java
	a) Difference between Compiler and Interpreter
4	Moving towards First Program of Java
	a) Description of main method
	b) How to download and install Java
	c) First Java program using Notepad,Edit plus and Eclipse IDE
	d) Command Line Argument
5	Token in Java
	a) Keywords, Identifiers, Literals,Punctuators and Operators
6	Types of Literals in java
	a) Integral Literal b) Floating point Literal c) Char Literal d) Boolean Literal e) String Literal f) null literal
7	Operators
	a) Arithmetic Operator b) Unary Operators c) Assignment Operator

	d) Relational Operator e) Logical Operators f) Boolean Operators g) Bitwise Operators h) Ternary Operator i) Member Operator(.) j) new Operator k) instanceof Operator
9	Programs on Method Parameter and return type
10	Introduction to Object OrientedProgramming
	a) OOPs Features and Advantages
	b) Class, Object, Abstraction, Encapsulation, Inheritance and Polymorphism
	c) Default constructor added bycompiler
	d) Why compiler adds default constructor to our class
	e) Types of variables (Primitive and Reference)
	f) Instance variable, Static variable,Parameter variable and local variable
	g) How to provide our own user defined values for instance variable
	h) this keyword
	i) Role of instance variable while creating the Object
	j) Working with static variable while creating the Object
	k) When we should declare a variable an instance or static variable?
	l) Data Hiding
	m) Encapsulation

	n) How to print object properties value(instance variable value)
	o) Setter and Getter
10	Introduction to Constructor
	a) Advantage of Constructor
	b) Types Of Constructor
	c) Default, No Argument and Parameterized Constructor
	d) Passing Object reference to the constructor(Copy Constructor)
	e) Instance Block in java
	f) How many ways we can initialize object properties (instance variable)
11	Relationship between the classes
	a) IS-A (Inheritance) Relation and HAS-A(Association) Relation
	b) Introduction to Inheritance (IS-Arelation)
	c) Types of Inheritance
	d) this() and super()
	e) Why java does not support multipleinheritance
	g) Access modifiers in Java
	g) HAS-A relation(Association)
	h) Composition and Aggregation
12	Wrapper classes in Java
	a) Autoboxing and Unboxing
13	Introduction to Polymorphism

	a) Method Overloading, Var-Args,
	b) Ambiguity issues while overloadinga method

	c) Method Overriding
	d) Upcasting and Downcasting
	e) @Override Annotation
	f) Role of Access Modifier whileOverriding a method
	g) Co-variant concept in methodoverriding
	h) Method Hiding
14	Final and Sealed keywords in Java
15	Object class and its method
	a) getClass(), hashCode(), toString(),equals(Object o), clone()
	b) wait(), notify(), notifyAll()finalize()
16	Inner classes in java
	a) Nested inner class, Method localinner class, static nested inner class, Anonymous inner class
17	Abstract class and abstract methods
18	Introduction to interface
	a) Default and static method(**Java 8 features**)
	b) Functional interfaces

	c) Lambda Expression
	c) Predicate<T>, Consumer<T>, Supplier<T>, Function<T,R>,BiPredicate<T,U>, BiConsumer<T,U>
	d) Marker interface e) Difference between Abstract class and Interface

19	Enum in java
	a) Writing enum inside the class, Outsideof the class and inside the method. b) Writing Constructor inside an enum. c) Passing enum in switch expression
20	JVM Architecture
	a) Class loader subsystem, RuntimeData areas and Execution Engine
	b) Different types of class loaders
	c) static block in Java
	d) ClassNotFoundException andNoClassDefFoundError
	e) Drawback of new keyword
	f) Method Area, Heap Memory, StackMemory, PC register, Native Method Stack.
	g) Garbage Collector
	h) Heap and Stack diagram Programs
	i) Execution Engine and JIT Compiler
21	Arrays in java
	a) 1-D Array, 2-D Array

	b) Multi-Dimensional Array
	c) Interview Standard Coding
	d) Arrays class methods
22	String Handling in Java
	a) String Immutability
	b) Various Methods of String class[20 Methods of String class]
	c) == operator and equals(Objectobj) method
	d) StringBuffer class and its method
	e) StringBuilder class and its method
	f) Performance Comparison of StringBuffer and StringBuilder
23	Exception Handling in Java
	a) Introduction to Exception
	b) Exception Hierarchy
	c) Different Criteria of Exception
	d) try-catch block
	e) Working with Generic and SpecificException
	f) Nested try catch, try with multicatch block (1.7)
	g) Dealing with Infinity and NaN
	h) Finally block
	i) Try with Resources (1.7)

	j) Exception Propagation
	k) Checked and Unchecked Exception
	l) throw and throws keyword
	m) User-defined checked andunchecked Exception
	n) Various Test cases with Checked and UncheckedException
	o) Remaining methods of Object classclone() and finalize()

23	Introduction to Multithreading
	a) Introduction to Process, Thread, Multitasking and multithreading
	b) Creating Thread by using Threadclass and Runnable interface
	c) Various methods of Thread class like start(), run(), isAlive(), sleep(long ms), join(), setName()getName(), currentThread(), setPriority(), getPriority()
	d) Implementation of Runnableinterface by using **Lambda Expression**
	e) Race condition in multithreading
	f) Synchronization (Method and blocklevel)
	g) Object and class levelSynchronization
	h) Thread life cycle
	i) Thread Group and Thread Pool

	j) Inter Thread Communication(ITC)
	k) Deadlock in multithreading
	l) interrupt() method of Thread class
	m) Daemon Thread in Java
24	Introduction to Java I/O Streams
	a) Introduction to Stream
	b) Serialization and De-Serialization
	c) transient keyword role in Serialization
25	Collection Framework
	a) Introduction to Collection Framework
	b) Collection using Legacy classes
	c) Collection Hierarchy
	d) Introduction to List, Set and QueueInterface
	e) Methods of Collections interface
	f) 9 ways to retrieve Collection objects including forEach() method (1.8), forEachRemaining() and MethodReference
	g) List interface introduction and Hierarchy
	h) List interface Methods
	i) Introduction to List implemented classes ArrayList, LinkedList, Vector and Stack
	j) ArrayList class with its methods
	k) LinkedList class with its methods

	l) Vector class and its Methods
	m) Stack class and its methods
	n) Introduction to Set interface and
	hashing technique
	o) Hierarchy of Set interface
	p) Introduction to Set interface and its Methods
	q) HashSet class with its methods
	r) LinkedHashSet class with Methods
	s) Introduction to SortedSet interface
	t) Comparable and Comparator interface
	u) TreeSet class with methods
	v) Methods of SortedSet interface
	w) Methods of NavigableSet interface
26	Introduction to Map interface
	a) Map interface Hierarchy
	b) Map interface Methods
	c) HashMap class with Methods
	d) LinkedHashMap class with methods
	e) Hashtable class with methods
	f) Properties class with methods
	g) IdentityHashMap class
	h) WeakHashMap class
	i) Introduction to SortedMap interface
	j) TreeMap class with methods

	k) Methods of SortedMap
	l) Methods of NavigableMap
	m) Introduction to Queue interface
	n) PriorityQueue class
27	Working with Generics
	Mixing generic and non-generic collections
	Polymorphism with Generic
	Type Erasure in Generic

	Wild Card in Generic<?>
28	Concurrent Collections in Java
	a) Limitation of Traditional Collection
	b) Synchronizing the TraditionalCollection
	c) ConcurrentModification in java
	d) CopyOnWriteArrayList class
	e) CopyOnWriteArraySet class
	f) ConcurrentHashMap class
29	Streams API
	a) Creation of Streams to process thedata
	b) Opertaion on Stream (Intermediateand Terminal)

	Intermediate Operation Methods: filter(Predicate<T> predicate), map(Function<T, R> mapper) , flatMap(Function<T, Stream<R>> mapper)distinct(), sorted(), sorted(Comparator<T> comparator),peek(Consumer<T> action), limit(long maxSize), skip(long n), takeWhile(Predicate<T> predicate), dropWhile(Predicate<T> predicate)
	Terminal Operation Methods: forEach(Consumer<T> action),toArray(), collect(Collector<T, A, R> collector), reduce(identity, BinaryOperator<T>accumulator), min(Comparator<T> comparator), max(Comparator<T> comparator),count(), anyMatch(Predicate<T> predicate), allMatch(Predicate<T> predicate), noneMatch(Predicate<T> predicate),findFirst() and
	findAny()
30	New Features of Java
	New Date and Time API, Optional class,Record class.