**Lesson 1:Introduction to Java**

 Responsiblities of Bytecode verifier,class loader,JIT compiler,JVM

**Lesson 3: Language Fundamentals**

 Discuss illegal assignments for e float ,char ,byte data types

 What happens when you ignore break in switch, Discuss the valid data types that w.r.t switch case

**Lesson 4: Classes and Objects**

 Discuss all the important points of enum

 What is the default value of all instance variables

 What will happen if you don’t initialize a local variable and try to print it?

 Garbage collector points

o Not assured even if you call System.gc() method ,least priority daemon thread,calls finalize method

 Static methods – discuss all the points

 What is created in heap?



**Lesson 5: Exploring Basic Java Class Libraries**

 What are wrapper classes list all the wrapper classes

 What are the possible modifiers for the top level class, instance variables , local variables

 (example final is the only modifier used for local variables)

 Which type of variables must be initialized-mandatory(ans:final variable)

 String ,stringbuffer and string builder – discuss which is mutable and methods ( append,concat ,etc),equlas and == w.r.t String ?

 Methods of object class- list out

 Scanner ,use delimiter method

 All new date features, LocalDate methods to get current date, tomorrows date, yesterdays

 Discuss equals and hashCode()

**Lesson 6: Inheritance and Polymorphism**

 Difference between overriding and overloading

 Abstract class,interfaces (modifiers of the data members in an interface)– discuss the points

 Aggregation relation ship – how will you implement in java

 Instanceof – discuss

 Discuss all points about key word “this” and “super”(while writing constructors)

 How will you write varargs (what conditions must be followed)

 All the points w.r.t final variable,method and class

IGATE Sensitive

**Lesson 7: Abstract Classes and Interfaces**

 By default interface data members are \_\_\_\_\_\_\_\_\_\_\_

**Lesson 9:Exception Handling**

 List all the checked exception and unchecked exception (discuss on classnotfound,classcastexception ,numberformatexception,sqlexception,ArrayIndexOutOfBoundException,NullPointerException,I OException)

 Base class of all exception

 How will you create checked and unchecked userdefined exception

 Try catch finally throw throws – all points

 Significance of Try-with-resource feature in exception handling

 Any null reference with method invocation will create null pointer exception example(very important)

o Example String var=null , s.op.(var.length())

 Difference between enhanced for loop and iterator

 Layered architecture with exception handling

**Lesson 13: File IO and Lesson 15: Property Files**

 Different types of streams in File IO,LineNumberReader,Buffered Streams,flush(),Serialization and Deserialization

 Below classes are in java.io package Reader,Write,InputStream,OutputStrean,FileInputStream,FileOutputStream,ObjectInputStream

 Discuss isFile()

**Lesson 16: Java Database Connectivity (JDBC 4.0)**

 What happens when rs.next is not given and we try to access rs.getXXX()

 setDate method of rs will accept only SQL date type

 in transaction management discuss about setAutoCommit – true and false, conn.commit(),con.rollBack()

 discuss all type drivers (Type1,2,3,4) and JDBC-ODBC Bridge Driver, Java to Database, Protocol, Java to Native API, Java to Network Protocol

 JDBC steps,What happens if the query does not return any result,

 Discuss execute,executeUpdate,executeQuery, Class.forName("oracle.jdbc.driver.OracleDriver");

 ResultSet methods,iterating resultset

 SQL Exception and all the scenarios when it occurs –( eg:what happens if the table does not exist)

 All points about resultset,all statement types

 Stored procedures using Preparecall statement

IGATE Sensitive

**Lesson 10: Array**

 Declare int array ,Boolean array syntax

**Lesson 11: Collection**

 Printing the collection using for loop and iterator.

 Linkedlist,Arraylist- all collections comparison for ordered,sorted,duplicates ,allows null

 Hashtable and vector are synchronized,

 SortedMap(entries are stored using Comparator,duplicate entries replace original entries,stored as key/value pair)

 Discuss- LinkedList,LinkedHashSet

 Collections.sort(),Arrays.sort(array),ways of iterating the collection, Diff between hashmap and hashtable (key/value pair, not sorted and not ordered)

 TreeSet discuss -(key/value pair,elements in the TreeSet should be of the type that implements comparable.Need to implement either Comparable or Comparator interface to sort user defined objects)

 Discuss clear(),removeAll(),isEmpty()

**Lesson 12:Generics**

 Use of Generics (introduced in JDK 1.5,used to avoid runtime exceptions like ClassCastException and casting)

**Lesson 14: Introduction to Junit 4 & Lesson 18: Advanced Testing**

 Explain @Test with all attributes like timeout ,expected…

 @ignore- explain

 Explain static import of Assert class

 Explain-'@RunWith(Suite.class) ,@Suite.SuiteClasses

 @Before,@After ,@BeforeClass,@AfterClass–explain

 What is parameterized test?

**Lesson 19: Logging with Log4J**

 What is logger configuration file . How many ways you can implement(XML and property file)

 Log4j API components – discuss

 Log4j levels –discuss

 What is Appender,root logger

 Logger configuration file format

**Lesson 20: Multithreading** IGATE Sensitive

 Thread API’s eg: static method to obtain the current thread,start(), run(),join()-waits for the other thread to terminate),Thread class constructors - discuss

 Thread Lifecycle (Thread States),Thread priority(integer values)

 2 ways of Creating thread.

 Wait(),notify and notifyAll() are in Object class

**Lesson 21: LambdaExpressions**

 Discuss Simple lambda expressions, how to write the functional interface

 Printing the list using lambda expression

**Lesson 22: Stream API**

 Consumer,BiFunctional,Predicate Functional interfaces – discuss with code snippets from the slides or solved examples

 Discuss below stream operations:

Array.stream (), Map, filter, forEach, collect,distinct,limit,reduce