**Lesson 1: Introduction to Java**

* Responsiblities of Bytecode verifier,class loader,JIT compiler,JVM
* **Byte code Verifier**: The byte code verifier ensures the following:
* the Java programs have been compiled correctly,
* they will obey the virtual machine’s access restrictions, and
* the byte codes will not access private data when they should not
* **Class Loader:**

When the loader retrieves classes from the network, it keeps classes from different servers separate from each other and from local classes. Through this separation, the class loader prevents a class that is loaded off the network from pretending to be one of the standard built-in classes, or from interfering with the operation of classes loaded from other servers.

**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

EXECUTED CONSECUTED CASES

BYTE SHORT INT CHAR ENUM STRING

**Lesson 4: Classes and Objects**

* Discuss all the important points of enum
* What is the default value of all instance variables

INT 0

DOUBLE FLOAT 0.0

STRING NULL

BOOLEAN FALSE

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

COMPILATION ERROR

* Garbage collector points
  + Not assured even if you call System.gc() method ,least priority daemon thread,calls finalize method

HINT JVM TO INVOKE GARBAGE COLLECTOR

* Static methods – discuss all the points

CAN BE INVOKE BY CLASS NAME

CAN ACCCESS OTHER STATIC MEMBERS ONLY

ONLY ONE COPY OF THAT TYPE

* 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 and public, private and static is used for local)
* 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 ?
* String IMMUTABLE (VALUES CAN NOT BE CHANGED)
* Array is IMMUTABLE
* stringbuffer and string builder- MUTABLE
* String Concatenation

String str=”Hello”;

str=str.concat(“Cap”);

* StringBuilder sb=new StringBuilder("abc”);

Sb.append(“def”);

* String str=”Hello”;

String str1=”Hello”;

SOP(str==str1)-----------🡪True bcoz both strings points to same reference in String Pull

\

* String str= new String(“Hello”);

String str1=”Hello”;

SOP(str=str1)---🡪False

* 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

The instanceOf operator compares an object to a specified type

Object o=new Worker ();

if(o instanceOf Worker)//🡪true

{Worker w=(Worker) 0;

}

* 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
* Can the final classes be instantiated,inherited ?
* Key words that can’t be used for final –abstract/extends

**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,IOException)

***Ioexception* *Classnotfoundexception, sqlexception,: checked***

***Classcastexception,numberformatexception, ArrayIndexOutOfBoundException,NullPointerException:Unchecked Exception***

* 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)
  + 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 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)
* Go through Comparator and Camparable interface methods
* Discuss clear(),removeAll(),isEmpty()

**Lesson 12:Generics**

* Use of Generics (introduced in JDK 1.5,used to avoid runtime exceptions like ClassCastException and casting)
* Why Generics(Is used to avoid runtime cast exceptions and it was introduced in JDK 1.5 version)

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

* Explain @Test with all attributes like timeout ,expected…
* @ignore- explain

IGNORE CASES

* Explain static import of Assert class
* Explain-'@RunWith(Suite.class) ,@Suite.SuiteClasses
* @Before,@After ,@BeforeClass,@AfterClass–explain

@Before,@After-BEFORE AND AFTER EACH TEST CASE

@BeforeClass,@AfterClass-MUST BE STATIC

* What is parameterized test?

**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
* Go through SQL Injection.Which interface is used to prevent SQL Injection
* Stored procedures using Preparecall statement
* Given the below procedure:
* PROCEDURE getNameById( id IN NUMBER, name OUT VARCHAR2)

The above stored procedure can be called as shown below

String sql = "{call getNameById(?, ?)}"

**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**

* 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
* Which exception is thrown by Sleep()

**Lesson 21: Lambda Expressions**

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

SAM-single Abstract Method

Annotation, or give only single function

* 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 (), list.stream() ,Map, filter, forEach(System.out::println),count() ,sorted(),distinct(),limit(),reduce**

* Which method used to implement parallel stream operation
* How to display lowest 3 values from a list by using stream methods . i.e stream().sorted().limit()

AFTER ADVICE

AFTER RETURN

AFTER THROWING

LIFE CYCLE OF BEAN