**Interview Questions:**

**Why Java is Better than other Programming Language?**

* **Java is platform independent:** Means Java can run on any computer irrespective to the hardware or software dependency. Means Java does not depend on hardware means what type of processor, RAM etc. Java will run on a machine which will satisfy its basic needs.
* **Java is a secure language:** Byte code concept separates java from all other language. As we know Byte code is set of symbols created by Sun Microsoft which are generated after the compilation of Program. This byte code is converted to machine code by JRE. This byte code is executed by the JRE which consist of byte code loader which loads byte code into memory and then byte code verifier which verifies the Byte code and look for any extra symbols and code which does not resembles the standards and throws unexpected code way and makes java byte code error free and secure.

Java is fast because of JIT compiler. Just In Time compiler stores the repeated code in its memory and in byte code where repeated code is used; instead of loading that code again from memory JIT use it from its cache memory and safe time and space and make execution fast.

* **“Compile Once and Run Forever”:** is famous tagline in Java and yes it is very true this is all possible because of byte code as once byte code is generated you can use that byte code and can run that program in any operating system, every operating system’s java runtime environment will convert that Byte code into machine code.

Due to its byte code is in Bytes, java program takes very less memory on the hard disk and therefore java technology is Portable also.

**What is a transient variable?**

A variable that may not be serialized.

**How are Observer and Observable used?**

Objects that subclass the Observable class maintain list of Observers. When an Observable object is updated it invokes update () method of each of its observers to notify that observers.

**Serialization in Java?**

It is a process in which current state of object will be saved in stream of bytes. As byte stream create is platform neutral once objects are created in one system can be deserialized in other platform.

**Use of Serialization:**

* Write to disk
* Store in memory
* Send byte stream to another platform over network
* Save byte stream into DB (As BLOB)

**Note:** Serializable Interface is a Marker Interface. Here there is no method in Serializable Interface.

* Serialization interface needs to be implemented in order to make object serialized.
* Transient instance variable doesn’t serialize with Object state.
* If Super class implements Serializable then subclasses are also Serializable automatically.
* If Super class is not Serializable then when subclass is de serialized the super class’s default constructor will be invoked. Hence all variable will get default value and reference will be null.

**Use of serialVersionUID:**

serialVerionUID is a version number associated to each Serializable class by serialization at runtime. This version number is used during deserialization process to verify that the sender and receiver of a serialized object have loaded class for that object which is compatible with respect to serialization.

* Defining serialVerionUID field in serializable class is not mandatory.
* If a serializable class have explicit serialVerionUID then this field should be of type long and must be static and final.
* If there is no serialVersionUID field defined explicitly then serialization at runtime will calculate default value for that class. Which can vary based on compiler implementation. Hence it is advisable to define serialVerionUID.
* If there is a difference between serialVerionUID of loader receiver class and corresponding sender class, then **InvalidClassException** will be thrown.

**What is Externalization?**

Externalization is not serialization but by implementing Externalizable interface to persist and restore the object. Custom Serialization process is used which is implemented by application.

Unlike Serializable interface, Externalizable interface is not marker interface and it provides two methods- writeExternal and readExternal. These methods are implemented by the class to give the class a complete control over the format.

How serialization happens? JVM first checks for Externalizable interface and if object supports Externalizable interface, then serializes the object using writeExternal method. If the object doesn’t support Externalizable but implements Serializable, then the object is saved using ObjectOutputStrem.

**Varargs?**

The varargs allow the method to accept zero or multiple arguments. Before varargs either we use overload method or take an array as the method parameter but it was not considered good because it leads to maintenance problem. If we don’t know how many arguments we will have to pass in the method, varargs are better approach.

* There can be only one variable argument in the method.
* Variable argument(varargs) must be last parameter.

**What is the difference between OOPs and Object based Programming?**

Object based programming follows OOPs except Inheritance, Encapsulation and Polymorphism.

**What is composition?**

Holding the reference of the other class within some other class is known as composition.

**What is difference between Aggregation and Composition?**

Aggregation represents weak relationship whereas composition represents strong relationship. For example: Bike has an indicator but bike has an engine.

**What is super in java?**

It is a keyword that refers to the immediate parent class object.

**Why method overloading is not possible by changing the return type in Java?**

Because of ambiguity.

**What is the difference between abstraction and encapsulation?**

Abstraction hides the implementation details where as encapsulation wraps code and data into single unit.

**Can you declare an interface method static?**

No, because methods of an interface are abstract by default, and static and abstract keywords can’t be used together.

**What is Exception Propagation?**

Forwarding the exception object to the invoking method is called exception Propagation.

**What is reflection?**

It is a process of modifying the runtime behavior of runtime class at runtime.

**What are wrapper classes?**

Wrapper classes that allow primitive types to be accessed as objects.

**What is native method?**

It is a method that is implemented in language other than Java.

**What is purpose of System class?**

It provides access to the system resources.

**What is JavaBean?**

Are reusable components written in Java Programming language, designed to be manipulated.

**What is shutdown hook?**

The shutdown hook is basically a thread i.e., invoked implicitly before JVM shuts down. So we can use it perform clean up resource.

**What is JDBC driver?**

JDBC driver is a software component that enables java application to interact with the database. There are 4 types of JDBC drivers:

* JDBC-ODBC driver
* Native-API driver (partially java driver)
* Network Protocol driver (fully java driver)
* Thin driver (Fully java driver)

**What are the steps to connect to the database in Java?**

* Registering the driver class

Class.forName(oracle.jdbc.driver.OracleDriver)

* Creating Connection

Connection con = DriverManager.getConnection(jdbc.oracle.thin:@localhost:1521:xe:”system”,”oracle”);

* Creating statement

Statement stmt = con.createStatement();

* Executing Queries

ReslutSet rs = stmt.executeQuery(“select \* from employee”);

* Closing Connection

Con.close();

**What is deep cloning() and shallow cloning()?**

**What is Internalization?**

It is a mechanism to create such an application that can be adapted to different languages and regions.

It is one of the powerful concept in Java if you are developing an application and want to display messages, currencies, date, time etc. according to the specific region or language.

Interview Questions for RESTFul webservices: