**Technical**

**Q1. What is the difference between an Inner Class and a Sub-Class?**

Ans: An Inner class is a class which is nested within another class. An Inner class has access rights for the class which is nesting it and it can access all variables and methods defined in the outer class.

A sub-class is a class which inherits from another class called super class. Sub-class can access all public and protected methods and fields of its super class.

**Q3. What’s the purpose of Static methods and static variables?**

Ans: When there is a requirement to share a method or a variable between multiple objects of a class instead of creating separate copies for each object, we use static keyword to make a method or variable shared for all objects.

**Q4. What is data encapsulation and what’s its significance?**

Ans: Encapsulation is a concept in Object Oriented Programming for combining properties and methods in a single unit.

Encapsulation helps programmers to follow a modular approach for software development as each object has its own set of methods and variables and serves its functions independent of other objects. Encapsulation also serves data hiding purpose.

**Q5. What is a singleton class? Give a practical example of its usage.**

A singleton class in java can have only one instance and hence all its methods and variables belong to just one instance. Singleton class concept is useful for the situations when there is a need to limit the number of objects for a class.

The best example of singleton usage scenario is when there is a limit of having only one connection to a database due to some driver limitations or because of any licensing issues.

**Q10. What is Final Keyword in Java? Give an example.**

Ans: In java, a constant is declared using the keyword Final. Value can be assigned only once and after assignment, value of a constant can’t be changed.

In below example, a constant with the name const\_val is declared and assigned avalue:

Private Final int const\_val=100

When a method is declared as final,it can NOT be overridden by the subclasses.This method are faster than any other method,because they are resolved at complied time.

When a class is declares as final,it cannot be subclassed. Example String,Integer and other wrapper classes.

**Q16. What are Java Packages? What’s the significance of packages?**

Ans: In Java, package is a collection of classes and interfaces which are bundled together as they are related to each other. Use of packages helps developers to modularize the code and group the code for proper re-use. Once code has been packaged in Packages, it can be imported in other classes and used.

**Q17. Can we declare a class as Abstract without having any abstract method?**

Ans: Yes we can create an abstract class by using abstract keyword before class name even if it doesn’t have any abstract method. However, if a class has even one abstract method, it must be declared as abstract otherwise it will give an error.

**Q18. What’s the difference between an Abstract Class and Interface in Java?**

Ans: The primary difference between an abstract class and interface is that an interface can only possess declaration of public static methods with no concrete implementation while an abstract class can have members with any access specifiers (public, private etc) with or without concrete implementation.

Another key difference in the use of abstract classes and interfaces is that a class which implements an interface must implement all the methods of the interface while a class which inherits from an abstract class doesn’t require implementation of all the methods of its super class.

A class can implement multiple interfaces but it can extend only one abstract class.

**Q23. How an object is serialized in java?**

Ans: In java, to convert an object into byte stream by serialization, an interface with the name Serializable is implemented by the class. All objects of a class implementing serializable interface get serialized and their state is saved in byte stream.

**Q24. When we should use serialization?**

Ans: Serialization is used when data needs to be transmitted over the network. Using serialization, object’s state is saved and converted into byte stream .The byte stream is transferred over the network and the object is re-created at destination.

**Q25. Is it compulsory for a Try Block to be followed by a Catch Block in Java for Exception handling?**

Ans: Try block needs to be followed by either Catch block or Finally block or both. Any exception thrown from try block needs to be either caught in the catch block or else any specific tasks to be performed before code abortion are put in the Finally block.

**Q26. Is there any way to skip Finally block of exception even if some exception occurs in the exception block?**

Ans: If an exception is raised in Try block, control passes to catch block if it exists otherwise to finally block. Finally block is always executed when an exception occurs and the only way to avoid execution of any statements in Finally block is by aborting the code forcibly by writing following line of code at the end of try block:

System.exit(0);

**Q29. Can we override static methods of a class?**

Ans: We cannot override static methods. Static methods belong to a class and not to individual objects and are resolved at the time of compilation (not at runtime).Even if we try to override static method,we will not get an complitaion error,nor the impact of overriding when running the code.

**Q31. Is String a data type in java?**

Ans: String is not a primitive data type in java. When a string is created in java, it’s actually an object of Java.Lang.String class that gets created. After creation of this string object, all built-in methods of String class can be used on the string object.

**Q33. Why Strings in Java are called as Immutable?**

Ans: In java, string objects are called immutable as once value has been assigned to a string, it can’t be changed and if changed, a new object is created.

In below example, reference str refers to a string object having value “Value one”.

String str="Value One";

When a new value is assigned to it, a new String object gets created and the reference is moved to the new object.

str="New Value";

**Q34. What’s the difference between an array and Vector?**

Ans: An array groups data of same primitive type and is static in nature while vectors are dynamic in nature and can hold data of different data types.

**Q36. Why Runnable Interface is used in Java?**

Ans: Runnable interface is used in java for implementing multi threaded applications. Java.Lang.Runnable interface is implemented by a class to support multi threading.

**Q37. What are the two ways of implementing multi-threading in Java?**

Ans: Multi threaded applications can be developed in Java by using any of the following two methodologies:

1. By using Java.Lang.Runnable Interface. Classes implement this interface to enable multi threading. There is a Run() method in this interface which is implemented.

2. By writing a class that extend Java.Lang.Thread class.

**Q72. What’s the difference between comparison done by equals method and == operator?**

Ans: In Java, equals() method is used to compare the contents of two string objects and returns true if the two have same value while == operator compares the references of two string objects.

In the following example, equals() returns true as the two string objects have same values. However == operator returns false as both string objects are referencing to different objects:

**OOPS**

1. **What is the abstraction in Java? (**[**answer**](https://javarevisited.blogspot.com/2010/10/abstraction-in-java.html#axzz6oOeSmpNw)**)**  
   An OOP technique to hide complexities from clients.
2. **What is Inheritance in Java? (**[**answer**](http://www.java67.com/2012/08/what-is-inheritance-in-java-oops-programming-example.html)**)**  
   An object-oriented technique to reuse code and functionalities.
3. **What is Encapsulation or data hiding in Java? (**[**answer**](https://javarevisited.blogspot.com/2012/03/what-is-encapsulation-in-java-and-oops.html)**)**  
   An oop way to hide data so that you can change it later without impacting others.

**8) Explain the use of abstraction in OOPs?**

* One of the main concepts of OOPs is Abstraction. It handles the program complexity and improves efficiency by hiding unnecessary information from the user, showing only essential attributes.
* Abstraction is selecting data from a large pool to show only relevant data of the object to the user.
* For example, you want to create a bank app and collect all your customer details.
* The customer details you come up with might include full name, address, contact, tax information, favorite place, etc. But only a few of them are required to create a bank app.
* Relevant data like name, address, etc., make sense for a banking application.
* Since we have fetched/selected/removed the customer information from a larger pool, the process is called abstraction in OOPs.

HIBERNATE

### ****2. What is ORM?****

Object Relational Mapping (ORM) is a programming technique that is used to convert data between relational databases and OOP (Object Oriented Programming) languages such as C#, Java, etc.

### ****3. What is the Java Persistence API (JPA)?****

Java Persistence API (also known as JPA) has the specification for managing the relational data in several applications. JPA’s current version 2.1 was started in July 2011 but approved as final on 22 May 2013. It also helps us in writing implementation-independent code.

### ****4. What is the major difference between Hibernate & JPA?****

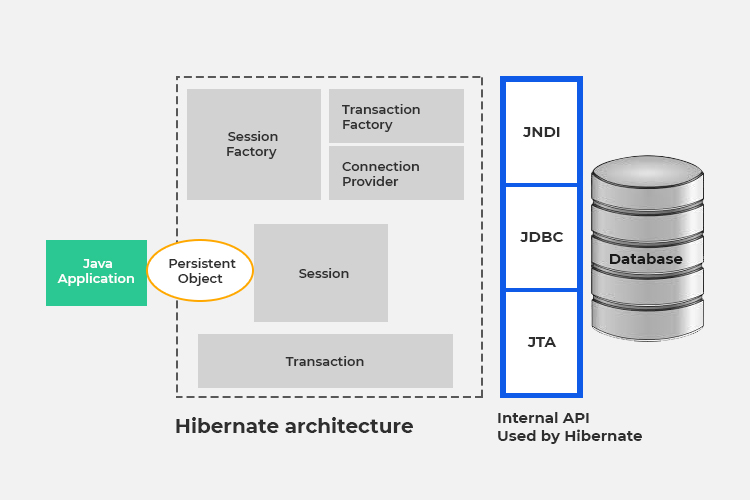
Hibernate is an ORM tool that is used to save the state of any java object into the database whereas JPA defines the management of relational data in the java applications.

**5. Explain Hibernate architecture?**

Hibernate architecture is based on many objects including session factory, persistent object, connection factory, transaction, session, transaction factory, etc. It is categorized into the following layers:

* Java application layer
* Hibernate framework layer
* Backhand API layer
* Database layer

As you can view in the diagram given below:



**6. What is HQL in Hibernate?**

Hibernate Query Language (HQL) is an object-oriented language that is independent of the database and considers the java objects in a similar way as that of the SQL.

**10. What are the main benefits of using the Hibernate Framework?**

Following are the main benefits of using the Hibernate Framework:

* We can focus on our business logic because it eliminates all the boiler-plate code that comes with JDBC and takes care of managing resources.
* It has powerful query language (HQL) which is based on object-oriented concepts and supports inheritance, polymorphism and association.
* Hibernate has great support for XML, and JPA annotations that make our code implementation independent.
* It is an open-source project and tons of online documentation are available over the internet.
* Hibernate supports lazy initialization and perform the database queries only when it is required.
* For database vendor-specific features, hibernate is best because we can execute native SQL queries.
* Hibernate cache helps to get better performance

**11. What are the key advantages of Hibernate over JDBC?**

Following are the key advantages of the hibernate framework over JDBC:

* Hibernate makes the code cleaner and more readable by removing boilerplate code.
* Unlike JDBC API, Hibernate supports inheritance, associations and collections.
* Hibernate provides transaction management while in JDBC API we need to write code for transaction management using commit and rollback.
* HQL is more object-oriented and very close to java programming language. For JDBC, it is mandatory to write native SQL queries.
* JDBC has poor performance because it doesn’t support caching but Hibernate has great caching options to provide best performance.
* Hibernate supports JPA annotations whereas JDBC code is tightly coupled with the application.

**22. How to map an entity to multiple tables?**

Following are the simple steps to map an entity to 2 database tables:

* Annotate your entity with JPA’s @Table and @SecondaryTable annotations, in addition, provide the names of the first and second table as the name parameters.
* Annotate each attribute that you need to map to the secondary table with a @Column annotation and add the secondary table name as the value of the table attribute.

**24. What are the collection types in hibernate?**

* Bag
* Set
* List
* Array
* Map

**27. What are the core interfaces of Hibernate?**

Following are the core interfaces of hibernate framework:

* Configuration
* SessionFactory
* Session
* Query
* Criteria
* Transaction