1) What is difference between JDK,JRE and JVM?

JVM

JVM is an acronym for Java Virtual Machine, it is an abstract machine which provides the runtime environment in which java bytecode can be executed. It is a specification.

JVMs are available for many hardware and software platforms (so JVM is platform dependent).

JRE

JRE stands for Java Runtime Environment. It is the implementation of JVM.

JDK

JDK is an acronym for Java Development Kit. It physically exists. It contains JRE + development tools.

2) How many types of memory areas are allocated by JVM?

Many types:

Class(Method) Area

Heap

Stack

Program Counter Register

Native Method Stack

3) What is JIT compiler?

Just-In-Time(JIT) compiler:It is used to improve the performance. JIT compiles parts of the byte code that have similar functionality at the same time, and hence reduces the amount of time needed for compilation.Here the term “compiler” refers to a translator from the instruction set of a Java virtual machine (JVM) to the instruction set of a specific CPU.

4) What is platform?

A platform is basically the hardware or software environment in which a program runs. There are two types of platforms software-based and hardware-based. Java provides software-based platform.

5) What is the main difference between Java platform and other platforms?

The Java platform differs from most other platforms in the sense that it's a software-based platform that runs on top of other hardware-based platforms.It has two components:

Runtime Environment

API(Application Programming Interface)

6) What gives Java its 'write once and run anywhere' nature?

The bytecode. Java is compiled to be a byte code which is the intermediate language between source code and machine code. This byte code is not platform specific and hence can be fed to any platform.

7) What is classloader?

The classloader is a subsystem of JVM that is used to load classes and interfaces.There are many types of classloaders e.g. Bootstrap classloader, Extension classloader, System classloader, Plugin classloader etc.

8) Is Empty .java file name a valid source file name?

Yes, save your java file by .java only, compile it by javac .java and run by java yourclassname Let's take a simple example:

9) Is delete,next,main,exit or null keyword in java?

No.

10) If I don't provide any arguments on the command line, then the String array of Main method will be empty or null?

It is empty. But not null.

11) What if I write static public void instead of public static void?

Program compiles and runs properly.

12) What is the default value of the local variables?

The local variables are not initialized to any default value, neither primitives nor object references.

13) What is difference between object oriented programming language and object based programming language?

Object based programming languages follow all the features of OOPs except Inheritance. Examples of object based programming languages are JavaScript, VBScript etc.

14) What will be the initial value of an object reference which is defined as an instance variable?

The object references are all initialized to null in Java.

Core Java - OOPs Concepts: Constructor Interview Questions

15) What is constructor?

Constructor is just like a method that is used to initialize the state of an object. It is invoked at the time of object creation.

16) What is the purpose of default constructor?

The default constructor provides the default values to the objects. The java compiler creates a default constructor only if there is no constructor in the class.

17) Does constructor return any value?

Ans:yes, that is current instance (You cannot use return type yet it returns a value).

18)Is constructor inherited?

No, constructor is not inherited.

19) Can you make a constructor final?

No, constructor can't be final.

Core Java - OOPs Concepts: static keyword Interview Questions

20) What is static variable?

static variable is used to refer the common property of all objects (that is not unique for each object) e.g. company name of employees,college name of students etc.

static variable gets memory only once in class area at the time of class loading.

21) What is static method?

A static method belongs to the class rather than object of a class.

A static method can be invoked without the need for creating an instance of a class.

static method can access static data member and can change the value of it.

22) Why main method is static?

because object is not required to call static method if It were non-static method,jvm creats object first then call main() method that will lead to the problem of extra memory allocation.

23) What is static block?

Is used to initialize the static data member.

It is excuted before main method at the time of classloading.

24) Can we execute a program without main() method?

Ans) Yes, one of the way is static block.

25) What if the static modifier is removed from the signature of the main method?

Program compiles. But at runtime throws an error "NoSuchMethodError".

26) What is difference between static (class) method and instance method?

static or class method   instance method

1)A method i.e. declared as static is known as static method.

A method i.e. not declared as static is known as instance method.

2)Object is not required to call static method.

Object is required to call instance methods.

3)Non-static (instance) members cannot be accessed in static context (static method, static block and static nested class) directly. static and non-static variables both can be accessed in instance methods.

4)For example: public static int cube(int n){ return n*n*n;}   For example: public void msg(){...}.

27) What is this in java?

It is a keyword that that refers to the current object.

28)What is Inheritance?

Inheritance is a mechanism in which one object acquires all the properties and behaviour of another object of another class. It represents IS-A relationship. It is used for Code Resusability and Method Overriding.

29) Which class is the superclass for every class.

Object class.

30) Why multiple inheritance is not supported in java?

To reduce the complexity and simplify the language, multiple inheritance is not supported in java in case of class.

31) What is composition?

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

32) What is difference between aggregation and composition?

Aggregation represents weak relationship whereas composition represents strong relationship. For example: bike has an indicator (aggregation) but bike has an engine (compostion).

33) Why Java does not support pointers?

Pointer is a variable that refers to the memory address. They are not used in java because they are unsafe(unsecured) and complex to understand.

34) What is super in java?

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

35) Can you use this() and super() both in a constructor?

No. Because super() or this() must be the first statement.

36)What is object cloning?

The object cloning is used to create the exact copy of an object.

37) What is method overloading?

If a class have multiple methods by same name but different parameters, it is known as Method Overloading. It increases the readability of the program.

38) Why method overloading is not possible by changing the return type in java?

Becauseof ambiguity.

39) Can we overload main() method?

Yes, You can have many main() methods in a class by overloading the main method.

40) What is method overriding:

If a subclass provides a specific implementation of a method that is already provided by its parent class, it is known as Method Overriding. It is used for runtime polymorphism and to provide the specific implementation of the method.

41) Can we override static method?

No, you can't override the static method because they are the part of class not object.

42) Why we cannot override static method?

It is because the static method is the part of class and it is bound with class whereas instance method is bound with object and static gets memory in class area and instance gets memory in heap.

43) Can we override the overloaded method?

Yes.

44) Difference between method Overloading and Overriding.

Method Overloading   Method Overriding

1) Method overloading increases the readability of the program.

Method overriding provides the specific implementation of the method that is already provided by its super class.

2) method overlaoding is occurs within the class.

Method overriding occurs in two classes that have IS-A relationship.

3) In this case, parameter must be different.	In this case, parameter must be same.

45) Can you have virtual functions in Java?

Yes, all functions in Java are virtual by default.

46) What is covariant return type?

Now, since java5, it is possible to override any method by changing the return type if the return type of the subclass overriding method is subclass type. It is known as covariant return type.

47) What is final variable?

If you make any variable as final, you cannot change the value of final variable(It will be constant).

48) What is final method?

Final methods can't be overriden.

49) What is final class?

Final class can't be inherited.

50) What is blank final variable?

A final variable, not initalized at the time of declaration, is known as blank final variable.

51) Can we intialize blank final variable?

Yes, only in constructor if it is non-static. If it is static blank final variable, it can be initialized only in the static block.

52) Can you declare the main method as final?

Yes, such as, public static final void main(String[] args){}.

53) What is Runtime Polymorphism?

Runtime polymorphism or dynamic method dispatch is a process in which a call to an overridden method is resolved at runtime rather than at compile-time.

In this process, an overridden method is called through the reference variable of a super class. The determination of the method to be called is based on the object being referred to by the reference variable.

54) Can you achieve Runtime Polymorphism by data members?

No.

56) What is abstraction?

Abstraction is a process of hiding the implementation details and showing only functionality to the user.

Abstraction lets you focus on what the object does instead of how it does it.

57) What is the difference between abstraction and encapsulation?

Abstraction hides the implementation details whereas encapsulation wraps code and data into a single unit.

58) What is abstract class?

A class that is declared as abstract is known as abstract class. It needs to be extended and its method implemented. It cannot be instantiated.

59) Can there be any abstract method without abstract class?

No, if there is any abstract method in a class, that class must be abstract.

60) Can you use abstract and final both with a method?

No, because abstract method needs to be overridden whereas you can't override final method.

61) Is it possible to instantiate the abstract class?

No, abstract class can never be instantiated.

62) What is interface?

Interface is a blueprint of a class that have static constants and abstract methods.It can be used to achieve fully abstraction and multiple inheritance.

63) Can you declare an interface method static?

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

64) Can an Interface be final?

No, because its implementation is provided by another class.

65) What is marker interface?

An interface that have no data member and method is known as a marker interface.For example Serializable, Cloneable etc.

66) What is difference between abstract class and interface?

Abstract class	Interface
1)An abstract class can have method body (non-abstract methods).	Interface have only abstract methods.
2)An abstract class can have instance variables.	An interface cannot have instance variables.
3)An abstract class can have constructor.	Interface cannot have constructor.
4)An abstract class can have static methods.	Interface cannot have static methods.
5)You can extends one abstract class.	You can implement multiple interfaces.

67) Can we define private and protected modifiers for variables in interfaces?

No, they are implicitly public.

68) When can an object reference be cast to an interface reference?

An object reference can be cast to an interface reference when the object implements the referenced interface.

69) What is package?

A package is a group of similar type of classes interfaces and sub-packages. It provides access protection and removes naming collision.

70) Do I need to import java.lang package any time? Why ?

No. It is by default loaded internally by the JVM.

71) Can I import same package/class twice? Will the JVM load the package twice at runtime?

One can import the same package or same class multiple times. Neither compiler nor JVM complains about it.But the JVM will internally load the class only once no matter how many times you import the same class.

72) What is static import ?

By static import, we can access the static members of a class directly, there is no to qualify it with the class name.

73) What is Exception Handling?

Exception Handling is a mechanism to handle runtime errors.It is mainly used to handle checked exceptions.

74) What is difference between Checked Exception and Unchecked Exception?

1)Checked Exception

The classes that extend Throwable class except RuntimeException and Error are known as checked exceptions e.g.IOException,SQLException etc. Checked exceptions are checked at compile-time.

2)Unchecked Exception

The classes that extend RuntimeException are known as unchecked exceptions e.g. ArithmeticException,NullPointerException etc. Unchecked exceptions are not checked at compile-time.

75) What is the base class for Error and Exception?

Throwable.

76) Is it necessary that each try block must be followed by a catch block?

It is not necessary that each try block must be followed by a catch block. It should be followed by either a catch block OR a finally block. And whatever exceptions are likely to be thrown should be declared in the throws clause of the method.

77) What is finally block?

finally block is a block that is always executed.

78) Can finally block be used without catch?

Yes, by try block. finally must be followed by either try or catch.

79) Is there any case when finally will not be executed?

finally block will not be executed if program exits(either by calling System.exit() or by causing a fatal error that causes the process to abort).

80) What is difference between throw and throws?

throw keyword	throws keyword
1)throw is used to explicitly throw an exception.	throws is used to declare an exception.
2)checked exceptions can not be propagated with throw only.	checked exception can be propagated with throws.
3)throw is followed by an instance.	throws is followed by class.
4)throw is used within the method.	throws is used with the method signature.
5)You cannot throw multiple exception	You can declare multiple exception e.g. public void method()throws IOException,SQLException.

81) Can an exception be rethrown?

Yes.

82) Can subclass overriding method declare an exception if parent class method doesn't throw an exception ?

Yes but only unchecked exception not checked.

83) What is exception propagation ?

Forwarding the exception object to the invoking method is known as exception propagation.

Java: String Handling Interview Questions

There is given a list of string handling interview questions with short and pointed answers. If you know any string handling interview question, kindly post it in the comment section.

84) What is the meaning of immutable in terms of String?

The simple meaning of immutable is unmodifiable or unchangeable. Once string object has been created, its value can't be changed.

85) Why string objects are immutable in java?

Because java uses the concept of string literal. Suppose there are 5 reference variables,all referes to one object "sachin".If one reference variable changes the value of the object, it will be affected to all the reference variables. That is why string objects are immutable in java.

86) How many ways we can create the string object?

There are two ways to create the string object, by string literal and by new keyword.

87) How many objects will be created in the following code?

String s1="Welcome";

String s2="Welcome";

String s3="Welcome";

Only one object.

88) Why java uses the concept of string literal?

To make Java more memory efficient (because no new objects are created if it exists already in string constant pool).

89)How many objects will be created in the following code?

String s = new String("Welcome");

Two objects, one in string constant pool and other in non-pool(heap).

90) What is the basic difference between string and stringbuffer object?

String is an immutable object. StringBuffer is a mutable object.

91) What is the difference between StringBuffer and StringBuilder ?

StringBuffer is synchronized whereas StringBuilder is not synchronized.

92) How can we create immutable class in java ?

We can create immutable class as the String class by defining final class and

93) What is the purpose of toString() method in java ?

The toString() method returns the string representation of any object. If you print any object, java compiler internally invokes the toString() method on the object. So overriding the toString() method, returns the desired output, it can be the state of an object etc. depends on your implementation.

Core Java : Nested classes and Interfaces Interview Questions

94)What is nested class?

A class which is declared inside another class is known as nested class. There are 4 types of nested class member inner class, local inner class, annonymous inner class and static nested class.

95) Is there any difference between nested classes and inner classes?

Yes, inner classes are non-static nested classes i.e. inner classes are the part of nested classes.

96) Can we access the non-final local variable, inside the local inner class?

No, local variable must be constant if you want to access it in local inner class.

97) What is nested interface ?

Any interface i.e. declared inside the interface or class, is known as nested interface. It is static by default.

98) Can a class have an interface?

Yes, it is known as nested interface.

99) Can an Interface have a class?

Yes, they are static implicitely.

117) What is Garbage Collection?

Garbage collection is a process of reclaiming the runtime unused objects.It is performed for memory management.

118) What is gc()?

gc() is a daemon thread.gc() method is defined in System class that is used to send request to JVM to perform garbage collection.

119) What is the purpose of finalize() method?

finalize() method is invoked just before the object is garbage collected.It is used to perform cleanup processing.

120) Can an unrefrenced objects be refrenced again?

Yes.

121)What kind of thread is the Garbage collector thread?

Daemon thread.

122)What is difference between final, finally and finalize?

final: final is a keyword, final can be variable, method or class.You, can't change the value of final variable, can't override final method, can't inherit final class.

finally: finally block is used in exception handling. finally block is always executed.

finalize():finalize() method is used in garbage collection.finalize() method is invoked just before the object is garbage collected.The finalize() method can be used to perform any cleanup processing.

123)What is the purpose of the Runtime class?

The purpose of the Runtime class is to provide access to the Java runtime system.

124)How will you invoke any external process in Java?

By Runtime.getRuntime().exec(?) method.

125)What is the difference between the Reader/Writer class hierarchy and the InputStream/OutputStream class hierarchy?

The Reader/Writer class hierarchy is character-oriented, and the InputStream/OutputStream class hierarchy is byte-oriented.

126)What an I/O filter?

An I/O filter is an object that reads from one stream and writes to another, usually altering the data in some way as it is passed from one stream to another.

127) What is serialization?

Serialization is a process of writing the state of an object into a byte stream.It is mainly used to travel object's state on the network.

128) What is Deserialization?

Deserialization is the process of reconstructing the object from the serialized state.It is the reverse operation of serialization.

129) What is transient keyword?

If you define any data member as transient,it will not be serialized.

130)What is Externalizable?

Externalizable interface is used to write the state of an object into a byte stream in compressed format.It is not a marker interface.

131)What is the difference between Serializalble and Externalizable interface?

Serializable is a marker interface but Externalizable is not a marker interface.When you use Serializable interface, your class is serialized automatically by default. But you can override writeObject() and readObject() two methods to control more complex object serailization process. When you use Externalizable interface, you have a complete control over your class's serialization process.

133) What is reflection?

Reflection is the process of examining or modifying the runtime behaviour of a class at runtime.It is used in:

IDE (Integreted Development Environment) e.g. Eclipse, MyEclipse, NetBeans.

Debugger

Test Tools etc.

134) Can you access the private method from outside the class?

Yes, by changing the runtime behaviour of a class if the class is not secured.

135)What is a JavaBean?

are reusable software components written in the Java programming language, designed to be manipulated visually by a software development environment, like JBuilder or VisualAge for Java.

---

1) What is multithreading?

Multithreading is a process of executing multiple threads simultaneously. Its main advantage is:

Threads share the same address space.

Thread is lightweight.

Cost of communication between process is low.

2) What is thread?

A thread is a lightweight subprocess.It is a separate path of execution.It is called separate path of execution because each thread runs in a separate stack frame.

3)What is the difference between preemptive scheduling and time slicing?

Under preemptive scheduling, the highest priority task executes until it enters the waiting or dead states or a higher priority task comes into existence. Under time slicing, a task executes for a predefined slice of time and then reenters the pool of ready tasks. The scheduler then determines which task should execute next, based on priority and other factors.

4) What does join() method?

The join() method waits for a thread to die. In other words, it causes the currently running threads to stop executing until the thread it joins with completes its task.

5) What is difference between wait() and sleep() method?

1) The wait() method is defined in Object class.

The sleep() method is defined in Thread class.

2) wait() method releases the lock.

The sleep() method doesn't releases the lock.

6) Is it possible to start a thread twice?

No, there is no possibility to start a thread twice. If we does, it throws an exception.

7) Can we call the run() method instead of start()?

yes, but it will not work as a thread rather it will work as a normal object so there will not be context-switching between the threads.

8) What about the daemon threads?

The daemon threads are basically the low priority threads that provides the background support to the user threads. It provides services to the user threads.

9)Can we make the user thread as daemon thread if thread is started?

No, if you do so, it will throw IllegalThreadStateException

10)What is shutdown hook?

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

11)When should we interrupt a thread?

We should interrupt a thread if we want to break out the sleep or wait state of a thread.

12) What is synchronization?

Synchronization is the capabilility of control the access of multiple threads to any shared resource.It is used:

To prevent thread interference.

To prevent consistency problem.

13) What is the purpose of Synchronized block?

Synchronized block is used to lock an object for any shared resource.

Scope of synchronized block is smaller than the method.

14)Can Java object be locked down for exclusive use by a given thread?

Yes. You can lock an object by putting it in a "synchronized" block. The locked object is inaccessible to any thread other than the one that explicitly claimed it.

15) What is static synchronization?

If you make any static method as synchronized, the lock will be on the class not on object.

16)What is the difference between notify() and notifyAll()?

The notify() is used to unblock one waiting thread whereas notifyAll() method is used to unblock all the threads in waiting state.

17)What is deadlock?

Deadlock is a situation when two threads are waiting on each other to release a resource. Each thread waiting for a resource which is held by the other waiting thread.

1) What is the difference between ArrayList and Vector?

No.	ArrayList	Vector
1)	ArrayList is not synchronized.	Vector is synchronized.
2)	ArrayList is not a legacy class.	Vector is a legacy class.
3)	ArrayList increases its size by 50% of the array size.	Vector increases its size by doubling the array size.

2) What is the difference between ArrayList and LinkedList?

No.	ArrayList	LinkedList
1)	ArrayList uses a dynamic array.	LinkedList uses doubly linked list.
2)	ArrayList is not efficient for manipulation because a lot of shifting is required.	LinkedList is efficient for manipulation.
3)	ArrayList is better to store and fetch data.	LinkedList is better to manipulate data.

3) What is the difference between Iterator and ListIterator?

Iterator traverses the elements in forward direction only whereas ListIterator traverses the elements in forward and backward direction.

No.	Iterator	ListIterator
1)	Iterator traverses the elements in forward direction only.	ListIterator traverses the elements in backward and forward directions both.
2)	Iterator can be used in List, Set and Queue.	ListIterator can be used in List only.

4) What is the difference between Iterator and Enumeration?

No.	Iterator	Enumeration

1)	Iterator can traverse legacy and non-legacy elements.	Enumeration can traverse only legacy elements.
2)	Iterator is fail-fast.	Enumeration is not fail-fast.
3)	Iterator is slower than Enumeration.	Enumeration is faster than Iterator.

5) What is the difference between List and Set?

List can contain duplicate elements whereas Set contains only unique elements.

6) What is the difference between HashSet and TreeSet?

HashSet maintains no order whereas TreeSet maintains ascending order.

7) What is the difference between Set and Map?

Set contains values only whereas Map contains key and values both.

8) What is the difference between HashSet and HashMap?

HashSet contains only values whereas HashMap contains entry(key,value). HashSet can be iterated but HashMap need to convert into Set to be iterated.

9) What is the difference between HashMap and TreeMap?

HashMap maintains no order but TreeMap maintains ascending order.

10) What is the difference between HashMap and Hashtable?

No.	HashMap	Hashtable
1)	HashMap is not synchronized.	Hashtable is synchronized.
2)	HashMap can contain one null key and multiple null values.	Hashtable cannot contain any null key or null value.

11) What is the difference between Collection and Collections?

Collection is an interface whereas Collections is a class. Collection interface provides normal functionality of data structure to List, Set and Queue. But, Collections class is to sort and synchronize collection elements.

12) What is the difference between Comparable and Comparator?

No.	Comparable	Comparator

1) Comparable provides only one sort of sequence. Comparator provides multiple sort of sequences.

2) It provides one method named compareTo(). It provides one method named compare().

3) It is found in java.lang package. it is found in java.util package.

4) If we implement Comparable interface, actual class is modified. Actual class is not modified.

13) What is the advantage of Properties file?

If you change the value in properties file, you don't need to recompile the java class. So, it makes the application easy to manage.

14) What does the hashCode() method?

The hashCode() method returns a hash code value (an integer number).

The hashCode() method returns the same integer number, if two keys (by calling equals() method) are same.

But, it is possible that two hash code numbers can have different or same keys.

15) Why we override equals() method?

The equals method is used to check whether two objects are same or not. It needs to be overridden if we want to check the objects based on property.

For example, Employee is a class that has 3 data members: id, name and salary. But, we want to check the equality of employee object on the basis of salary. Then, we need to override the equals() method.

16) How to synchronize List, Set and Map elements?

Yes, Collections class provides methods to make List, Set or Map elements as synchronized:

17) What is the advantage of generic collection?

If we use generic class, we don't need typecasting. It is typesafe and checked at compile time.

18) What is hash-collision in Hashtable and how it is handled in Java?

Two different keys with the same hash value is known as hash-collision. Two different entries will be kept in a single hash bucket to avoid the collision.

19) What is the Dictionary class?

The Dictionary class provides the capability to store key-value pairs.

20) What is the default size of load factor in hashing based collection?

The default size of load factor is 0.75. The default capacity is computed as initial capacity * load factor. For example, 16 * 0.75 = 12. So, 12 is the default capacity of Map.

1) What is JDBC?

JDBC is a Java API that is used to connect and execute query to the database. JDBC API uses jdbc drivers to connects to the database.

2) 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 bridge driver

Native-API driver (partially java driver)

Network Protocol driver (fully java driver)

Thin driver (fully java driver)

3) What are the steps to connect to the database in java?

Registering the driver class

Creating connection

Creating statement

Executing queries

Closing connection

4) What are the JDBC API components?

The java.sql package contains interfaces and classes for JDBC API.

Interfaces:

Connection

Statement

PreparedStatement

ResultSet

ResultSetMetaData

DatabaseMetaData

CallableStatement etc.

Classes:

DriverManager

Blob

Clob

Types

SQLException etc.

5) What are the JDBC statements?

There are 3 JDBC statements.

Statement

PreparedStatement

CallableStatement

6) What is the difference between Statement and PreparedStatement interface?

In case of Statement, query is complied each time whereas in case of PreparedStatement, query is complied only once. So performance of PreparedStatement is better than Statement.

7) How can we execute stored procedures and functions?

By using Callable statement interface, we can execute procedures and functions.

8) What is the role of JDBC DriverManager class?

The DriverManager class manages the registered drivers. It can be used to register and unregister drivers. It provides factory method that returns the instance of Connection.

9) What does the JDBC Connection interface?

The Connection interface maintains a session with the database. It can be used for transaction management. It provides factory methods that returns the instance of Statement, PreparedStatement, CallableStatement and DatabaseMetaData.

10) What does the JDBC ResultSet interface?

The ResultSet object represents a row of a table. It can be used to change the cursor pointer and get the information from the database.

11) What does the JDBC ResultSetMetaData interface?

The ResultSetMetaData interface returns the information of table such as total number of columns, column name, column type etc.

12) What does the JDBC DatabaseMetaData interface?

The DatabaseMetaData interface returns the information of the database such as username, driver name, driver version, number of tables, number of views etc.

13) Which interface is responsible for transaction management in JDBC?

The Connection interface provides methods for transaction management such as commit(), rollback() etc.

14) What is batch processing and how to perform batch processing in JDBC?

By using batch processing technique in JDBC, we can execute multiple queries. It makes the performance fast.

15) How can we store and retrieve images from the database?

By using PreparedStatement interface, we can store and retrieve images.