Java interview questions

1. Why Java is a platform-independent language?

   Java language was developed in such a way that it does not depend on any hardware or software due to the fact that the compiler compiles the code and then converts it to platform-independent byte code which can be run on multiple systems. The only condition to run that byte code is for the machine to have a runtime environment (JRE) installed in it.

   For example,

   You can compile a program in Windows and can run it in Unix

2. Why is java not a pure object-oriented programming language?

   Java supports primitive data types - byte, boolean, char, short, int, float, long, and double and hence it is not a pure **object-oriented language**.

3. Difference between Heap and Stack Memory in java?

   Stack memory is the portion of memory that was assigned to every individual program. And it was fixed. On the other hand, Heap memory is the portion that was not allocated to the java program but it will be available for use by the java program when it is required, mostly during the runtime of the program.

4. Can we override static method in java?

No, you cannot override static method in java. You can only hide them.

5. Can you overload main method in java?

   Yes, you can overload main method in java but when you run your program,JVM will search for public static void main(String[] args) and execute that method.

6. What is the base class for all the classes?

   The super base class of all the Java classes is the java. lang. Object class

7. Can we override private methods in java?

   You cannot override a private or static method in Java. If you create a similar method with same return type and same method arguments in child class then it will hide the super class method; this is known as method hiding. Similarly, you cannot override a private method in sub class because it's not accessible there.

8. What do you mean by data encapsulation?

   - Data Encapsulation is an Object-Oriented Programming concept of hiding the data attributes and their behaviours in a single unit.
   - It helps developers to follow modularity while developing software by ensuring that each object is independent of other objects by having its own methods, attributes, and functionalities.
   - t is used for the security of the private properties of an object and hence serves the purpose of data hiding.

9. Can you list down some of important method from object class?

Important methods of object classes are:

- hashcode : It returns hash value of the object
- equals : It compares the object references
- wait : It causes current thread to wait until notify or notifyAll is not called
- notify : Wakes up single thread which is waiting for lock
- notifyAll: Wakes up all threads which is waiting for lock
- toString : Provides String representation of the object
- clone : This method is used to clone the object
- finalize: This method is called when object is being garbage collected

10. Which two methods should you override while putting the custom object as Key in HashMap?

- You need to override hashcode and equals method in custom class while putting objects of custom class in HashMap.

11. What is the difference between HashMap and HashSet in java?

- HashMap vs HashSet:

• Parameter	• HashMap	• HashSet
• Interface	• This is core difference among them.HashMap implements Map interface	• HashSet implement Set interface
• Method for storing data	• It stores data in a form of	• It uses add(value)

	key->value pair.So it uses put(key,value) method for storing data	method for storing data
• Duplicates	• HashMap allows duplicate value but not duplicate keys	• HashSet does not allow duplicate values.
• Performance	• It is faster than hashset as values are stored with unique keys	• It is slower than HashMap
• HashCode Calculation	• In hash map hashcode value is calculated using key object	• In this,hashcode is calculated on the basis of value object.Hashcode can be same for two value object so we have to implement equals() method.If equals() method return false then two objects are different.

12. Can we have an abstract class without having any abstract method in it?

    Yes, we can have an abstract class without Abstract Methods as both are independent concepts. Declaring a class abstract means that it can not be instantiated on its own and can only be sub-classed. Declaring a method abstract means that the Method will be defined in the subclass.

13. Have you heard about transient variables? When will you use it?

    Transient is a variables modifier used in serialization. At the time of serialization, if we don't want to save the value of a particular variable in a file, then we use the transient keyword. When JVM comes across a transient keyword, it ignores the original value of the variable and saves the default value of that variable data type

14. Difference between throw and throws keywords in Java?

    The throws keyword is used to declare which exceptions can be thrown from a method, while the throw keyword is used to explicitly throw an exception within a method or block of code.

15. Why Strings in java is immutable?

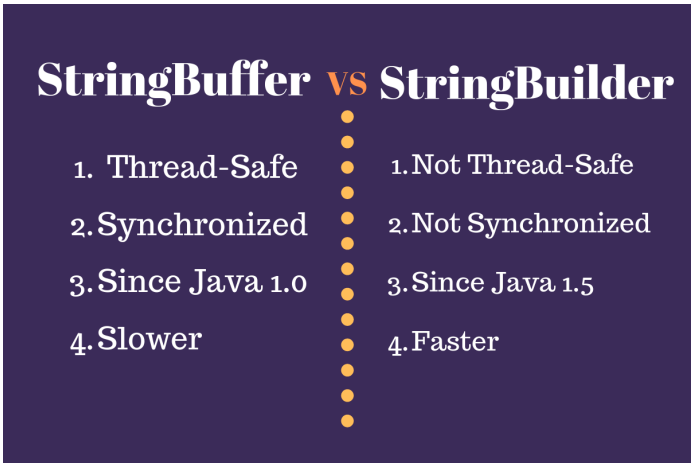
    The string is immutable means that we cannot change the object itself, but we can change the reference to the object. The string is made final to not allow others to extend it and destroy its immutability.

16. Can we have a static method in the interface?

- Till Java 7 we can't have static methods in interfaces.

- From the 1.8 version onwards in addition to default methods, we can write static methods also inside the interface to define utility functions.
- Interface static methods by default not available to the implementation classes hence by using implementation class reference we can't call interface static methods. We should call interface static methods by using the interface name.

17. What is the difference between StringBuffer and StringBuilder?



18. Can you declare the constructor as final?

    No, we cannot make constructors as final in java.

19. Difference between this() and super() in Java?

    The this() in the constructor refers to the current class object. The super() in the constructor refers immediate parent class object.

20. What is volatile in java?

For Java, “volatile” tells the compiler that the value of a variable must never be cached as its value may change outside of the scope of the program itself.

21. What are two different ways to call a garbage collector?

- There are 2 ways to call the garbage collector in java.
- You can use the Runtime. getRuntime(). gc() method- This class allows the program to interface with the Java Virtual machine. The “gc()” method allows us to call the garbage collector method.
- You can also use the System. gc() method which is common.

22. How many objects will be created below:

```
String str1= new String("John");

String str2= new String("John");
```

Though we are creating 2 instances, in the memory of String Pool we will have only one instance of it create with 2 references pointing to it in Constant Pool

23. Can you differentiate between a Checked Exception and an Unchecked exception?

A checked exception is caught at compile time whereas a runtime or unchecked exception is, as it states, at runtime. A checked exception

must be handled either by re-throwing or with a try-catch block, whereas an unchecked isn't required to be handled.

24. What is the difference between ArrayList and LinkedList? How will you decide which one you need to use?

ArrayList vs LinkedList:

Parameter	ArrayList	LinkedList
Internal data structure	It uses dynamic array to store elements internally	It uses doubly Linked List to store elements internally
Manipulation	If We need to insert or delete element in ArrayList, it may take O(n), as it internally uses array and we may have to shift elements in case of insertion or deletion	If We need to insert or delete element in LinkedList, it will take O(1), as it internally uses doubly LinkedList
Search	Search is faster in ArrayList as uses array internally which is index based. So here time complexity is O(1)	Search is slower in LinkedList as uses doubly Linked List internally So here time complexity is O(n)
Interfaces	ArrayList implements List interface only, So it	LinkedList implements List,Deque interfaces,

	can be used as List only	so it can be used as List,Stack or Queue

ArrayList provides constant time for search operation, so it is better to use ArrayList if searching is more frequent operation than add and remove operation. The LinkedList provides constant time for add and remove operations. So it is better to use LinkedList for manipulation.

26. You have started three threads from main threads. You need to make sure main thread complete last. How will you do it?

Using join method:

27. What do you mean by Platform independence of java?

You can write and compile program in one Operating system and run in other operating system.

For example:

You can compile program in Windows and can run it in Unix.

28. What is difference between JVM, JRE and JDK ?

JVM : JVM stands for Java Virtual Machine. It is virtual machine which actually runs the byte code.

JRE : JRE stands for Java Runtime Environment. It provides runtime environment for java code. It has JVM , libraries such as rt.jar and other files.
JDK : JDK stands for Java development kit. It is superset of JRE, it has JRE + compilation and debugging tools(javac and java).

29. What are memory areas allocated in JVM?
Memory areas allocated in JVM are:

- Heap area
- Method area
- JVM language stacks
- Program counter (PC) register
- Native method stacks

30. What is the difference between Class and Objects in Java

A class defines object properties including a valid range of values, and a default value.

A class also describes object behavior.

An object is a member or an "instance" of a class.

An object has a state in which all of its properties have values that you either explicitly define or that are defined by default settings.

31. What is Abstraction?

Data abstraction is the process of hiding certain details and showing only essential information to the user. Abstraction can be achieved with either abstract classes or interfaces

32. What is Encapsulation?

Encapsulation in Java is the process by which data (variables) and the code that acts upon them (methods) are integrated as a single unit. By encapsulating a class's variables, other classes cannot access them, and only the methods of the class can access them.

Data Hiding: Data Encapsulation

Access specifier is always private: Access

33. What is Polymorphism in java?

Polymorphism refers to the ability of a class to provide different implementations of a method, depending on the type of object that is passed to the method. To put it simply, polymorphism in Java allows us to perform the same action in many different ways.

34. What is Inheritance in java?

Inheritance is a mechanism wherein a new class is derived from an existing class. In Java, classes may inherit or acquire the properties and methods of other classes. A class derived from another class is called a subclass, whereas the class from which a subclass is derived is called a superclass

35. What is Constructor in java?

A constructor in Java is a special method that is used to initialize objects. The constructor is called when an object of a class is created.

36. Why main() in java is declared as public static void main? What if the main method is declared as
private?

Since the main method in Java is not supposed to return any value, it's made void which simply means main is not returning anything. Summary:
1. The main method must be declared public, static and void in Java otherwise, JVM will not able to run Java program.

37. What is immutable object in java?

The immutable objects are objects whose value can not be changed after initialization. We can not change anything once the object is created. For example, primitive objects such as int, long, float, double, all legacy

classes, Wrapper class, String class, etc. In a nutshell, immutable means unmodified or unchangeable.

38. What are access modifier available in java?

Java provides four types of access modifiers or visibility specifiers i.e. default, public, private, and protected

39. What is the difference between Abstract class and interface?

Abstract class vs Interface

Parameter	**Abstract class**	**Interface**
Default method Implementation	It can have default method implementation	Interfaces are pure abstraction.It can not have implementation at all but in java 8, you can have default methods in interface.
Implementation	Subclasses use **extends** keyword to extend an abstract class and they need to provide implementation of all the declared methods in the abstract class unless the	subclasses use **implements** keyw ord to implement interfaces and should provide implementation for all the methods declared in the interface

	subclass is also an abstract class	
Constructor	Abstract class can have constructor	Interface can not have constructor
Different from normal java class	Abstract classes are almost same as java classes except you can not instantiate it.	Interfaces are altogether different type
Access Modifier	Abstract class methods can have public ,protected,private and default modifier	Interface methods are by default public. you can not use any other access modifier with it
Main() method	Abstract classes can have main method so we can run it	Interface do not have main method so we can not run it.
Multiple inheritance	Abstract class can extends one other class and can implement one or more interface.	Interface can extends to one or more interfaces only
Speed	It is faster than interface	Interface is somewhat slower as it takes some time to find implemented method in class

Adding new method	If you add new method to abstract class, you can provide default implementation of it. So you don't need to change your current code	If you add new method to interface, you have to change the classes which are implementing that interface

40. Can one interface implement another interface in java?
No, One interface can not implement another interface. It can extend it using extends keyword.

41. What is marker interface?
Marker interfaces are interfaces which have no method but it is used to indicate JVM to behave specially when any class implement these interfaces.

For example : If you implement cloneable interface and then call .clone method of object, it will clone your object. If you do not implement cloneable interface, it will throw cloneNotSupported exception.

42. What is method overloading and method overriding in java?

Method overloading : Method overloading is concept that allows a class to have same method name but diferent method arguments. Method overloading is also known as compile time polymorphism.

Method overriding : If child class contain same method as parent class with same method signature. This is called method overriding. Method overriding is also known as dynamic polymorphism.

43. Can you override static methods in Java?

No, you can not override static methods in Java. You can create same method in child class but it won't be dynamic polymorphism. It will be method hiding. Static methods belong at class level not at object level hence you can not override static method

44. What is Enum in java?

Java Enum is special data type which represents list of constants values. It is a special type of java class. It can contain constant, methods and constructors etc.

45. Can we have try without catch block in java?

Yes, we can have try without catch block by using finally block. You can use try with finally. As you know finally block always executes even if you have exception or return statement in try block except in case of System.exit().

46. What are ways to create a thread in java?

There are two ways to create a thread in java

- By extending thread class
- By implementing the Runnable interface.

47. What are differences between Sleep and wait in java?

Parameter	**wait**	**sleep**
Synchonized	wait should be called from synchronized context i.e. from block or method, If you do not call it using synchronized context, it will throw IllegalMonitorStateException	It need not be called from synchronized block or methods
Calls on	wait method operates on Object and defined in Object class	Sleep method operates on current thread and is in java.lang.Thread
Release of lock	wait release lock of object on which it is called and also other locks if it holds any	Sleep method does not release lock at all
Wake up condition	until call notify() or notifyAll() from Object class	Until time expires or calls interrupt()
static	wait is non static method	sleep is static method

48. Define states of thread in java?

There are 5 states of thread in java

New : When you create a thread object and it is not alive yet.

Runnable: When you call start method of thread, it goes into Runnable state. Whether it will execute immediately or execute after some times , depends on thread scheduler.

Running : When thread is being executed, it goes to running state.

Blocked : When thread waits for some resources or some other thread to complete (due to thread's join), it goes to blocked state.

Dead: When thread's run method returns, thread goes to dead state.

49. Can we call run method directly to start a thread?

No, you can not directly call the run method to start a thread. You need to call the start method to create a new thread. If you call the run method directly, it won't create a new thread and it will be in the same stack as the main.

50. What is garbage Collection?

Garbage Collection is a process of looking at heap memory and deleting unused object present in heap memory. Garbage Collection frees unused memory. Garbage Collection is done by JVM.

51. What is System.gc()?

This method is used to invoke garbage collection for clean up unreachable object but it is not guaranteed that when you invoke System.gc() , garbage collection will definitely trigger.

52. What is use of finalize() method in object class?

Finalize method get called when object is being collected by Garbage Collector. This method can be used to write clean code before object is collected by Garbage Collector.

53. What is difference between final, finally and finalize in Java?

final : Final is a keyword which is used with class to avoid being extended, with instance variable so they can not reassigned, with methods so that they can not be overridden.

finally : Finally is a keyword used with try, catch and finally blocks. Finally block executes even if there is an exception. It is generally used to do some clean up work.

finalize : Finalize is a method is used to invoke garbage collection for clean up unreachable object but it is not guaranteed that when you invoke System.gc(), garbage collection will definitely trigger.

54. How do you prevent a class from being sub-classed in Java?

You can prevent a class from being subclassed by **using the final keyword in the class's declaration**. Similarly, you can prevent a method from being overridden by subclasses by declaring it as a final method. An abstract class can only be subclassed; it cannot be instantiated.

55. Will below program compile:

```
package com.java;

interface Decorable {
    String color="Blue";
}
public class Room implements Decorable{

   public static void main(String[] args) {
      System.out.println("Decorating room with Color: "+Room.color);
   }
```

}

Ans :
Yes, it will be compile.
Interface‘s variables are by default public static final ans as Room implements decorable, you will be able to access color variable of decorable interface.

56. What is the use of Classloader in Java?

A Java program is made up of a different number of custom classes and pre-defined classes. When a program is executed, JVM is used to load all the content of that needed class and through the use of Classloader JVM, it finds that class.

There are three types of Classloaders:

- **System Class Loader**
  It loads all the classes from the classpath.
- **Extension ClassLoader**
  It loads all the classes from the extension directory.
- **Bootstrap Class Loader**
  It loads all the pre-defined java classes.

57. Which class is a superclass of all classes?

Java.lang.The object is the root class for all the java classes and we don’t need to extend it. Every other java classes fall back under the object. All

the different non-primitive types including arrays are inherited directly or indirectly from this class.

58. What is the static keyword?

The static keyword is used with a class level variable to make it global so all the objects will be able to share the same variable. It can also be used with methods. A static method can access only static variables of the class and invoke only a static method of the class.

59. What is Type casting in Java?

Casting in Java is one of the top topics from where you can get questions in your interview. When we assign a value of one data type to a different data type then these two data types might not be compatible with each other and needs conversion. If data types are compatible with each other like, in case of the conversion of int value to long then automatic conversion is done by Java and doesn't require typecasting. But if data types are not compatible with each other then they need to be cast for conversion.

Syntax

dataType variablename = (dataType) variableToConvert;

60. What is the inner and anonymous inner class?

- In Java, we can define a class inside a class and they are called nested classes. Any nested class which is non-static are known as inner class. Inner classes are associated with objects of the class and they can access all the variables and methods of the outer class.

- Any local inner class without any name is known as an anonymous inner class. It is defined and instantiated in a single statement. Anonymous inner class always extend a class or implement an interface. Since an anonymous inner class doesn't have any name, it is not possible to create its constructor.

61. What is break and continue statement?

- In a while or do-while loop, we use break for a statement to terminate the loop. We use a break statement in a switch statement to exit the switch case. We can also use break statement for terminating the nested loop.
- The continue statement is used for skipping the current iteration of a for, while or do-while loop. We can use the break statement with a label to skip the current iteration of the outermost loop.

62. What is aggregation in Java?

Aggregation is best defined as the entity reference where it represents the relationship between two classes where the aggregate class contains a reference to the class which it owns. Aggregation represents a has-a and whole/part relationship.

63. What is the use of System class in Java?

- Java System class is one of the core classes. One of the easiest ways to log information for debugging is System.out.print() method. System class is final so we can't subclass and override its behavior through inheritance.
- System class doesn't provide any public constructors, so we can't instantiate this class and that's why all of its methods are static. Some of the utility methods of System class are for array copy, get the current time, and reading environment variables.

64. What is an instanceof keyword?

We can use instanceof keyword in java to check whether an object belongs to a class or not. We should avoid much usage of it.

65. What is an Iterator?

Iterator interface provides methods to iterate over any collection. We can get iterator instance from a collection using iterator() method. Iterator takes the place of Enumeration in the Java Collection Framework. The iterator allows the caller to remove elements from the underlying collection during the iteration.

66. What is the Java Collections Framework?

- Collections are used in every programming language and when initial java was released it contained few classes for collections: Vector, Stack, Array, and Hashtable. But for larger scope and usage, Java 1.2 came up with Collection Framework that grouped all the collections interfaces, implementations, and algorithms.

- Java Collection has come a long way with the usage of Generic and concurrent Collection classes for thread-safe operations. It has included blocking interfaces and their implementations in Java concurrent package.

67. What do you understand about Thread Priority?

Every thread when gets born is assigned with a priority value and usually higher priority gets precedence in execution but it also depends on the Thread Scheduler implementation which is OS dependent. We can assign the priority of thread but it doesn't guarantee that higher priority will get executed before lower priority thread. Thread priority is an integer value varies from 1 to 10 where 1 is the lowest and 10 is the highest priority thread.

68. What is Thread Scheduler and Time Slicing?

- Thread Scheduler is Operating System service which allocates the CPU time to the available runnable threads. Once a thread is created and it's in the runnable phase then its execution depends on the implementation of the Thread Scheduler.

- Time Slicing is a process of dividing available CPU time among the various runnable threads. Allocation of CPU time will depend on the thread priority or for how much time it is in the waiting state for getting the CPU time. Thread Scheduling cannot be controlled by Java, so it's always better to control it by the application itself.

69. Which is more preferred – Synchronized method or Synchronized block?

The synchronized block is more preferred because it doesn't lock the object, synchronized methods lock the object and if there are multiple synchronization blocks in the class, even though they are not related, it will stop the execution and put them in a wait state to get the lock on the object.

70. How to create daemon thread in Java?

Thread class setDaemon(true) is used for creating daemon thread in Java. We used to call this method before calling the start() method else it will give IllegalThreadStateException.

71. What is ThreadLocal?

ThreadLocal in Java is used for creating thread-local variables. We know that all threads of an object share its variables. So, if the variable is not threaded safe then we can use synchronization. But if we want to avoid synchronization then we can use ThreadLocal variables.

72. What happens when an exception is thrown by the main method?

When an exception is thrown by the main() method, Java Runtime terminates the program and print the exception message and stack trace in system console.

73. What is an Object in Java?

- An object in Java is a data structure that represents a real-world entity. In Java, an object can be a physical object like a car, or it can be an abstract concept like a mathematical formula.

- Each object has its own data and behavior. Data is the information that the object contains, while behavior is the object's ability to perform certain actions.
- Java objects are created using a class. A class is a template that defines the data and behavior of a particular type of object. Once a class has been defined, we can create objects of that class by using the new keyword.

74. How to get the database server details in Java program?

For this, we can use DatabaseMetaData object to get the database server details. When the database connection is created successfully, we can get the metadata object by calling getMetaData() method. There are also many methods in DatabaseMetaData that we can use to know the product name, it's version and configuration details.

```
1. DatabaseMetaData metadata = con.getMetaData();
2. String dbProduct = metadata.getdatabaseProductName;
```

75. What is JDBC PreparedStatement?

JDBC PreparedStatement object represents a precompiled SQL statement. We can use its setter method to set the variables for the query. Since PreparedStatement is precompiled, it can then be used to efficiently execute this statement multiple times. PreparedStatement is a better choice than Statement because it automatically escapes the special characters and avoids SQL injection attacks.

76. What is JIT compiler?

Just-In-Time(JIT) compiler: It is used to improve the performance. JIT compiles parts of the bytecode 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.

77. What if I write static public void instead of public static void?

The program compiles and runs correctly because the order of specifiers doesn't matter in Java.

78. What is the default value of the local variables?

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

79. What are the advantages of Packages in Java?

- Packages avoid the name clashes.
- The Package provides easier access control.
- We can also have the hidden classes that are not visible outside and used by the package
- It is easier to locate the related classes.

80. How many types of constructors are used in Java?

- **Default Constructor:** default constructor is the one which does not accept any value. The default constructor is mainly used to initialize

the instance variable with the default values. It can also be used for performing some useful task on object creation. A default constructor is invoked implicitly by the compiler if there is no constructor defined in the class.

- **Parameterized Constructor:** The parameterized constructor is the one which can initialize the instance variables with the given values. In other words, we can say that the constructors which can accept the arguments are called parameterized constructors.

81. Can we overload the constructors?

Yes, the constructors can be overloaded by changing the number of arguments accepted by the constructor or by changing the data type of the parameters.

82. Can we execute a program without main() method?

No, It was possible before JDK 1.7 using the static block. Since JDK 1.7, it is not possible.

83. Why is multiple inheritance not supported in java?

To reduce the complexity and simplify the language, multiple inheritance is not supported in java. Consider a scenario where A, B, and C are three classes. The C class inherits A and B classes. If A and B classes have the same method and you call it from child class object, there will be ambiguity to call the method of A or B class. Since the compile-time errors are better than runtime errors, Java renders compile-time error if you inherit 2

classes. So whether you have the same method or different, there will be a compile time error.

84. Why does Java not support pointers?

The 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.

85. What is object cloning?

The object cloning is used to create the exact copy of an object. The clone() method of the Object class is used to clone an object.
The **java.lang.Cloneable** interface must be implemented by the class whose object clone we want to create. If we don't implement Cloneable interface, clone() method generates CloneNotSupportedException.

```
protected Object clone() throws CloneNotSupportedException
```

86. Can we overload the main() method?

Yes, we can have any number of main methods in a Java program by using method overloading.

87. Can we change the scope of the overridden method in the subclass?

Yes, we can change the scope of the overridden method in the subclass. However, we must notice that we cannot decrease the accessibility of the method. The following point must be taken care of while changing the accessibility of the method.

- The private can be changed to protected, public, or default.

- The protected can be changed to public or default.
- The default can be changed to public.
- The public will always remain public.

88. Can you declare the main method as final?

Yes, We can declare the main method as public static final void main(String[] args){}.

89. What is JDBC in Java?

JDBC(Java Database Connectivity) is a Java API, which is helpful in interaction with the database to retrieve, manipulate and process the data using SQL. It will make use of JDBC drivers for connecting with the database. By using JDBC, we can access tabular data stored in various types of relational databases such as Oracle, MySQL, MS Access, etc.

90. What is JDBC driver?

- JDBC driver is a software component having various classes and interfaces, that
  enables the Java application to interact with a database.
- To connect with individual databases, JDBC requires particular drivers for each
  specific database. These drivers are provided by the database vendor in addition to the database

91. What are the JDBC statements?

In JDBC, Statements are used to send SQL commands to the database and receive data from the database. There are various methods provided by JDBC statements such as execute(), executeUpdate(), executeQuery, etc. which helps you to interact with the database.

There is three type of JDBC statements given in the following table.

- Statement: Statement is the factory for resultset. It is used for general purpose access to the database. It executes a static SQL query at runtime.
- PreparedStatement: The PreparedStatement is used when we need to provide input parameters to the query at runtime.
- CallableStatement: CallableStatement is used when we need to access the database stored procedures. It can also accept runtime parameters.

92. What are the functions of 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 return the instance of Statement, PreparedStatement, CallableStatement, and DatabaseMetaData..

93. What is Rowset?

A RowSet is an object that encapsulates a row set from either JDBC result sets or
tabular data sources such as files or spreadsheets. It supports component-based

development models like JavaBeans, with the help of a standard set of properties and event notifications. A RowSet is an object that encapsulates a row set from either JDBC result sets or tabular data sources such as files or spreadsheets. It supports component-based development models like JavaBeans, with the help of a standard set of properties and event notifications.

94. Which JDBC driver is fastest and used more commonly?

JDBC Net pure Java driver(Type 4 driver) is the fastest driver for localhost and remote connections because it directly interacts with the database by converting the JDBC call into vendor-specific protocol calls.

96. Explain JDBC API components.

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

Interfaces:

- Connection: The object of Connection is created by using the getConnection() method of DriverManager class. DriverManager is the factory for connection.
- Statement: The object of the Statement is created by using the createStatement()

method of the Connection class. The Connection interface is the factory for Statement.

- PreparedStatement: The PreparedStatement object is created by using prepareStatement() method of the Connection class. It is used for executing the parameterized query.
- ResultSet: The ResultSet object maintains a cursor pointing to a table row. At first, the cursor points before the first row. The executeQuery() method of the Statement interface returns the object of ResultSet.

Classes:

- DriverManager: It pretends to be an interface between the user and drivers.
  DriverManager keeps track of the available drivers and handles establishing a
  connection between a database and the relevant driver. It contains various
  methods to keep the interaction between the user and drivers**.**

99. Explain the usage of the getter and setter methods in ResultSet.

- **Getter methods:** These are used for retrieving the particular column values of the table from ResultSet. As a parameter, either the column index value or column name should be passed. Usually, the getter method is represented as getXXX() methods.
  Example: int getInt(string Column_Name)
  The above statement is used to retrieve the value of the specified column Index and the return type is an int data type.

- **Setter Methods:** These methods are used to set the value in the database. It is almost similar to getter methods, but here it requires to

pass the data/values for the particular column to insert into the database and the column name or index value of that column. Usually, setter method is represented as setXXX() methods.
Example: void setInt(int Column_Index, int Data_Value)
The above statement is used to insert the value of the specified column Index with an int value.