Viva Question generally asked in Viva

Differences between Abstraction and Encapsulation

Abstraction and encapsulation are complementary concepts. On the one hand, abstraction focuses on the behavior of an object. On the other hand, encapsulation focuses on the implementation of an object's behavior. Encapsulation is usually achieved by hiding information about the internal state of an object and thus, can be seen as a strategy used in order to provide abstraction.

What is JVM ? Why is Java called the Platform Independent Programming Language?

A Java virtual machine (JVM) is a process virtual machine that can execute Java bytecode. Each Java source file is compiled into a bytecode file, which is executed by the JVM. Java was designed to allow application programs to be built that could be run on any platform, without having to be rewritten or recompiled by the programmer for each separate platform. A Java virtual machine makes this possible, because it is aware of the specific instruction lengths and other particularities of the underlying hardware platform.

What is the Difference between JDK and JRE ?

The Java Runtime Environment (JRE) is basically the Java Virtual Machine (JVM) where your Java programs are being executed. It also includes browser plugins for applet execution. The Java Development Kit (JDK) is the full featured Software Development Kit for Java, including the JRE, the compilers and tools (like JavaDoc, and Java Debugger), in order for a user to develop, compile
and execute Java applications.

What does the "static" keyword mean ? Can you override private or static method in Java ?

The static keyword denotes that a member variable or method can be accessed, without requiring an instantiation of the class to which it belongs. A user cannot override static methods in Java, because method overriding is based upon dynamic binding at runtime and static methods are statically binded at compile time. A static method is not associated with any instance of a class so the concept is not applicable.

Can you access non static variable in static context ?

A static variable in Java belongs to its class and its value remains the same for all its instances. A static variable is initialized when the class is loaded by the JVM. If your code tries to access a non-static variable, without any instance, the compiler will complain, because those variables are not created yet and they are not associated with any instance.

What is Function Overriding and Overloading in Java ?
Method overloading in Java occurs when two or more methods in the same class have the exact same name, but different parameters. On the other hand, method overriding is defined as the case when a child class redefines the same method as a parent class. Overridden methods must have the same name, argument list, and return type. The overriding method may not limit the access of the method it overrides.

What is a Constructor, Constructor Overloading in Java and Copy-Constructor
A constructor gets invoked when a new object is created. Every class has a constructor. In case the programmer does not provide a constructor for a class, the Java compiler (Javac) creates a default constructor for that class. The constructor overloading is similar to method overloading in Java. Different constructors can be created for a single class. Each constructor must have its own unique parameter list. Finally, Java does support copy constructors like C++, but the difference lies in the fact that Java
doesn't create a default copy constructor if you don't write your own.
what value is a variable of the String type automatically initialized?
Answer: The default value of a String type is null.

Does Java support multiple inheritance ?
No, Java does not support multiple inheritance. Each class is able to extend only on one class, but is able to implement more than one interfaces.

Why Java doesn't support multiple inheritances?
Answer: Because of "Diamond Problem", Java doesn't support multiple inheritances in classes.

Why Java is not a pure Object Oriented language?
Answer: Java supports primitive types such as int, byte, short, long, etc that why it is not said to be a pure object-oriented language.

What are the access modifiers?
Answer: Java provides three access controls such as public, private and protected access modifier. When none of these are used, it's called default access modifier.

Can we overload the main method?
Answer:
Yes, we can overload the main method with syntax as public static void main(String args[]).

What access modifiers can be used for class?
Answer: We can use only two access modifiers for class public and default.

public: A class with a public modifier can be visible
1) In the same class
2) In the same package subclass
3) In the same package nonsubclass
4) In the different package subclass
5) In the different package nonsubclass.
default: A class with default modifier can be accessed
1) In the same class
2) In the same package subclass
3) In the same package nonsubclass
4) In the different package subclass
5) In the different package nonsubclass. ( )

Can we use catch statement for checked exceptions?
Answer: If there is no chance of raising an exception in our code then we can't declare catch block for handling
checked exceptions. This raises a compile-time error if we try to handle checked exceptions when there is
no possibility of causing an exception.

36. Explain a situation where finally block will not be executed?
Answer: Finally, the block will not be executed whenever JVM shutdowns. If we use system.exit(0) in try statement finally block if present will not be executed.

37. Explain about the main() method in java?
Answer: The main () method is the starting point of execution for all java applications.
public static void main(String[] args) {}
String args[] are an array of string objects we need to pass from command line arguments.
Every Java application must have at least one main method.

What is constructor in java?
Answer: A constructor is a special method used to initialize objects in the java. we use constructors to initialize all variables in the class when an object is created. As and when an object
is created it is initialized automatically with the help of constructor in java.
We have two types of constructors
Default Constructor
Parameterized Constructor

How can we find the actual size of an object on the heap?
Answer: In Java, there is no way to find out the actual size of an object on the heap.

In how many ways we can do synchronization in java?
Answer:
There are two ways to do synchronization in java:
1) Synchronized methods
2) Synchronized blocks
To do synchronization we use the synchronized keyword. (apex training)

Explain about Automatic type conversion in java?
Answer:
Java automatic type conversion is done if the following conditions are met:
1) When two types are compatible
Ex: int, float
int can be assigned directly to float variable.
2) Destination type is larger than source type.
Ex: int, long.
Int can be assigned directly to long .Automatic type conversion takes place if int is assigned to long
because long is larger datatype than int.
Widening Conversion comes under Automatic type conversion.

In how many ways we can do exception handling in java?
Answer:
We can handle exceptions in either of the two ways :
1) By specifying a try-catch block where we can catch the exception.
2) Declaring a method with throws clause.

What does null mean in java?
Answer:
When a reference variable doesn't point to any value it is assigned null.
Example: Employee employee;
In the above example employee object is not instantiate so it is pointed nowhere.

Can we define a package statement after the import statement in java?
Answer: We can't define a package statement after the import statement in java. a package statement must be the first
statement in the source file. We can have commented before the package statement.

What is the difference between an Interface and an Abstract class ?
Java provides and supports the creation both of abstract classes and interfaces. Both implementations share some common characteristics, but they differ in the following features:

- All methods in an interface are implicitly abstract. On the other hand, an abstract class may contain both abstract and nonabstract methods.

- A class may implement a number of Interfaces, but can extend only one abstract class.

- In order for a class to implement an interface, it must implement all its declared methods. However, a class may not implement all declared methods of an abstract class. Though, in this case, the sub-class must also be declared as abstract.

- Abstract classes can implement interfaces without even providing the implementation of interface methods.

- Variables declared in a Java interface is by default final. An abstract class may contain non-final variables.

- Members of a Java interface are public by default. A member of an abstract class can either be private, protected or public.

- An interface is absolutely abstract and cannot be instantiated. An abstract class also cannot be instantiated, but can be invoked if it contains a main method.

Does Importing a package imports its sub-packages as well in Java?

Ans: In java, when a package is imported, its sub-packages aren't imported and developer needs to import them separately if required.

For example, if a developer imports a package university.*, all classes in the package named university are loaded but no classes from the sub-package are loaded. To load the classes from its sub-package (say department), developer has to import it explicitly as follows:

Import university.department.*

Q21. Can we declare the main method of our class as private?

Ans: In java, main method must be public static in order to run any application correctly. If main method is declared as private, developer won't get any compilation error however, it will not get executed and will give a runtime error.

Q22. How can we pass argument to a function by reference instead of pass by value?

Ans: In java, we can pass argument to a function only by value and not by reference.

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.

Q32. In the below example, how many String Objects are created?

```
String s1="I am Java Expert";

String s2="I am C Expert";

String s3="I am Java Expert";
```

Ans: In the above example, two objects of Java.Lang.String class are created. s1 and s3 are references to same 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.

Q35. What is multi-threading?

Ans: Multi threading is a programming concept to run multiple tasks in a concurrent manner within a single program. Threads share same process stack and running in parallel. It helps in performance improvement of any program.

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.

Q40. How garbage collection is done in Java?

Ans: In java, when an object is not referenced any more, garbage collection takes place and the object is destroyed automatically. For automatic garbage collection java calls either System.gc() method or Runtime.gc() method.

Q41. How we can execute any code even before main method?

Ans: If we want to execute any statements before even creation of objects at load time of class, we can use a static block of code in the class. Any statements inside this static block of code will get executed once at the time of loading the class even before creation of objects in the main method.

Q42. Can a class be a super class and a sub-class at the same time? Give example.

Ans: If there is a hierarchy of inheritance used, a class can be a super class for another class and a sub-class for another one at the same time.

In the example below, continent class is sub-class of world class and it's super class of country class.

Q45. Can we call the constructor of a class more than once for an object?

Ans: Constructor is called automatically when we create an object using new keyword. It's called only once for an object at the time of object creation and hence, we can't invoke the constructor again for an object after its creation.

Q46. There are two classes named classA and classB. Both classes are in the same package. Can a private member of classA can be accessed by an object of classB?

Ans: Private members of a class aren't accessible outside the scope of that class and any other class even in the same package can't access them.

Q47. Can we have two methods in a class with the same name?

Ans: We can define two methods in a class with the same name but with different number/type of parameters. Which method is to get invoked will depend upon the parameters passed.

Q49. What's the benefit of using inheritance?

Ans: Key benefit of using inheritance is reusability of code as inheritance enables sub-classes to reuse the code of its super class. Polymorphism (Extensibility) is another great benefit which allow new functionality to be introduced without effecting existing derived classes.

Q50. What's the default access specifier for variables and methods of a class?

Ans: Default access specifier for variables and method is package protected i.e variables and class is available to any other class but in the same package,not outside the package.

Q59. Can we use a default constructor of a class even if an explicit constructor is defined?

Ans: Java provides a default no argument constructor if no explicit constructor is defined in a Java class. But if an explicit constructor has been defined, default constructor can't be invoked and developer can use only those constructors which are defined in the class.

Q60. Can we override a method by using same method name and arguments but different return types?

Ans: The basic condition of method overriding is that method name, arguments as well as return type must be exactly same as is that of the method being overridden. Hence using a different return type doesn't override a method.

Q61. A person says that he compiled a java class successfully without even having a main method in it? Is it possible?

Ans: main method is an entry point of Java class and is required for execution of the program however; a class gets compiled successfully even if it doesn't have a main method. It can't be run though.

Q62. Can we call a non-static method from inside a static method?

Ans: Non-Static methods are owned by objects of a class and have object level scope and in order to call the non-Static methods from a static block (like from a static main method), an object of the class needs to be created first. Then using object reference, these methods can be invoked.

Q66. Can a constructor have different name than a Class name in Java?

Ans: Constructor in Java must have same name as the class name and if the name is different, it doesn't act as a constructor and compiler thinks of it as a normal method.

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:

Q73. Is it possible to define a method in Java class but provide it's implementation in the code of another language like C?

Ans: Yes, we can do this by use of native methods. In case of native method based development, we define public static methods in our Java class without its implementation and then implementation is done in another language like C separately.

Q78. Is it correct to say that due to garbage collection feature in Java, a java program never goes out of memory?

Ans: Even though automatic garbage collection is provided by Java, it doesn't ensure that a Java program will not go out of memory as there is a possibility that creation of Java objects is being done at a faster pace compared to garbage collection resulting in filling of all the available memory resources.

So, garbage collection helps in reducing the chances of a program going out of memory but it doesn't ensure that.

Q83. How can an exception be thrown manually by a programmer?

Ans: In order to throw an exception in a block of code manually, **throw** keyword is used.

Q84. I want my class to be developed in such a way that no other class (even derived class) can create its objects. How can I do so?

Ans: If we declare the constructor of a class as private, it will not be accessible by any other class and hence, no other class will be able to instantiate it and formation of its object will be limited to itself only.

Q88. What happens if an exception is not handled in a program?

Ans: If an exception is not handled in a program using try catch blocks, program gets aborted and no statement executes after the statement which caused exception throwing.

Q89. I have multiple constructors defined in a class. Is it possible to call a constructor from another constructor's body?

Ans: If a class has multiple constructors, it's possible to call one constructor from the body of another one using **this()**.

Q90. What's meant by anonymous class?

Ans: An anonymous class is a class defined without any name in a single line of code using new keyword.

Q92. If an application has multiple classes in it, is it okay to have a main method in more than one class?

Ans: If there is main method in more than one classes in a java application, it won't cause any issue as entry point for any application will be a specific class and code will start from the main method of that particular class only.

Q95. String and StringBuffer both represent String objects. Can we compare String and StringBuffer in Java?

Ans: Although String and StringBuffer both represent String objects, we can't compare them with each other and if we try to compare them, we get an error.

Q96. Which API is provided by Java for operations on set of objects?

Ans: Java provides a Collection API which provides many useful methods which can be applied on a set of objects. Some of the important classes provided by Collection API include ArrayList, HashMap, TreeSet and TreeMap.

Q97. Can we cast any other type to Boolean Type with type casting?

Ans: No, we can neither cast any other primitive type to Boolean data type nor can cast Boolean data type to any other primitive data type.

Q98. Can we use different return types for methods when overridden?

Ans: The basic requirement of method overriding in Java is that the overridden method should have same name, and parameters.But a method can be overridden with a different return type as long as the new return type extends the original.

Q100. What's the order of call of constructors in inheritance?

Ans: In case of inheritance, when a new object of a derived class is created, first the constructor of the super class is invoked and then the constructor of the derived class is invoked.

What are the two types of Exceptions in Java ? Which are the differences between them ?
Java has two types of exceptions: checked exceptions and unchecked exceptions. Unchecked exceptions do not need to be declared in a method or a constructor's throws clause, if they can be thrown by the execution of the method or the constructor, and propagate outside the method or constructor boundary. On the other hand, checked exceptions must be declared in a method or a constructor's throws clause. See here for tips on Java exception handling.

What is a native method?
Answer: A native method is a method that is applied in a language other than Java.(

What value does read Line() return when it has reached the end of a file?
Answer:
The readLine() method returns null when it has reached the end of a file.

Can we create a constructor in abstract class?
Answer: We can create a constructor in the abstract class, it doesn't give any compilation error. But when we cannot
instantiate class there is no use in creating a constructor for abstract class.

What are abstract methods in java?
Answer: An abstract method is a method which doesn't have anybody. An abstract method is declared with
keyword abstract and semicolon in place of the method body.
Signature : public abstract void ();
Ex : public abstract void get details();
It is the responsibility of subclass to provide implementation to an abstract method defined in the abstract class.

6.2 What is the difference between Exception and Error in java ?
Exception and Error classes are both subclasses of the Throwable class. The Exception class is used for exceptional conditions that a user's program should catch. The Error class defines exceptions that are not excepted to be caught by the user program.

6.3 What is the difference between throw and throws ?
The throw keyword is used to explicitly raise a exception within the program. On the contrary, the throws clause is used to indicate those exceptions that are not handled by a method. Each method must explicitly specify which exceptions does not handle, so the callers of that method can guard against possible exceptions. Finally, multiple exceptions are separated by a comma.

6.4 What is the importance of finally block in exception handling ?
A finally block will always be executed, whether or not an exception is actually thrown. Even in the case where the catch statement is missing and an exception is thrown, the finally block will still be executed. Last thing to mention is that the finally block is used to release resources like I/O buffers, database connections, etc.

3.1 What is the difference between processes and threads ?
A process is an execution of a program, while a Thread is a single execution sequence within a process. A process can contain multiple threads. A Thread is sometimes called a lightweight process.

3.2 Explain different ways of creating a thread. Which one would you prefer and why ?

There are three ways that can be used in order for a Thread to be created:
• A class may extend the Thread class.
• A class may implement the Runnable interface.
• An application can use the Executor framework, in order to create a thread pool.
The Runnable interface is preferred, as it does not require an object to inherit the Thread class. In case your application design requires multiple inheritance, only interfaces can help you. Also, the thread pool is very efficient and can be implemented and used very easily.

Explain the available thread states in a high-level.
During its execution, a thread can reside in one of the following states:
• Runnable: A thread becomes ready to run, but does not necessarily start running immediately.

• Running: The processor is actively executing the thread code.
• Waiting: A thread is in a blocked state waiting for some external processing to finish.
• Sleeping: The thread is forced to sleep.
• Blocked on I/O: Waiting for an I/O operation to complete.
• Blocked on Synchronization: Waiting to acquire a lock.
• Dead: The thread has finished its execution.

What is synchronization and why is it important?
Answer: The term synchronization is the ability to control the access of multiple threads to shared resources. And it is important because, without it, it is not possible for one thread to modify a shared object while another thread is in the process of using or updating that object's value. This often leads to major errors.

What is the difference between a synchronized method and a synchronized block ?
In Java programming, each object has a lock. A thread can acquire the lock for an object by using the synchronized keyword. The synchronized keyword can be applied in a method level (coarse grained lock) or block level of code (fine grained lock).

How does thread synchronization occurs inside a monitor ? What levels of synchronization can you apply ?
The JVM uses locks in conjunction with monitors. A monitor is basically a guardian that watches over a sequence of synchronized code and ensuring that only one thread at a time executes a synchronized piece of code. Each monitor is associated with an object reference. The thread is not allowed to execute the code until it obtains the lock.

3.6 What's a deadlock ?
A condition that occurs when two processes are waiting for each other to complete, before proceeding. The result is that both processes wait endlessly.

3.7 How do you ensure that N threads can access N resources without deadlock ? A very simple way to avoid deadlock while using N threads is to impose an ordering on the locks and force each thread to follow that ordering. Thus, if all threads lock and unlock the mutexes in the same order, no deadlocks can arise.

4.1 What are the basic interfaces of Java Collections Framework ?
Java Collections Framework provides a well designed set of interfaces and classes that support operations on a collections of objects. The most basic interfaces that reside in the Java Collections Framework are:
• Collection, which represents a group of objects known as its elements.

• Set, which is a collection that cannot contain duplicate elements.
• List, which is an ordered collection and can contain duplicate elements.
• Map, which is an object that maps keys to values and cannot contain duplicate keys.

4.3 What is an Iterator ?

The Iterator interface provides a number of methods that are able to iterate over any Collection. Each Java Collection contains
the iterator method that returns an Iterator instance. Iterators are capable of removing elements from the underlying collection
during the iteration.

4.6 How HashMap works in Java ?

A HashMap in Java stores key-value pairs. The HashMap requires a hash function and uses hashCode and equals methods,
in order to put and retrieve elements to and from the collection respectively. When the put method is invoked, the HashMap calculates the hash value of the key and stores the pair in the appropriate index inside the collection. If the key exists, its value is updated with the new value. Some important characteristics of a HashMap are its capacity, its load factor and the threshold resizing.

4.10 What is difference between ArrayList and LinkedList ?

Both the ArrayList and LinkedList classes implement the List interface, but they differ on the following features:
• An ArrayList is an index based data structure backed by an Array. It provides random access to its elements with a performance equal to O(1). On the other hand, a LinkedList stores its data as list of elements and every element is linked to its previous and next element. In this case, the search operation for an element has execution time equal to O(n).
• The Insertion, addition and removal operations of an element are faster in a LinkedList compared to an ArrayList, because
there is no need of resizing an array or updating the index when an element is added in some arbitrary position inside the collection.
• A LinkedList consumes more memory than an ArrayList, because every node in a LinkedList stores two references, one for its previous element and one for its next element.
Check also our article ArrayList vs. LinkedList.

5.1 What is the purpose of garbage collection in Java, and when is it used ?

The purpose of garbage collection is to identify and discard those objects that are no longer needed by the application, in order for the resources to be reclaimed and reused.

5.2 What does System.gc() and Runtime.gc() methods do ?
These methods can be used as a hint to the JVM, in order to start a garbage collection. However, this it is up to the Java Virtual Machine (JVM) to start the garbage collection immediately or later in time.

5.3 When is the finalize() called ? What is the purpose of finalization ?
The finalize method is called by the garbage collector, just before releasing the object's memory. It is normally advised to release resources held by the object inside the finalize method.

When does an Object becomes eligible for Garbage collection in Java ?
A Java object is subject to garbage collection when it becomes unreachable to the program in which it is currently used.