

gulniaslan@gmail.com

Interface support multiple implementations.	Abstract class does not support multiple inheritance.
Interface does not contain Data Member.	Abstract class contains Data Member.

An interface does not contain constructor	Abstract class contains constructor.
An Interface Contains only incomplete member (signature of member)	An abstract class Contains both incomplete (abstract) and complete member
An interface cannot have access modifiers by default everything is assumed as public	An abstract class can contain access modifiers for the subs, functions, properties.
Member of interface can not be static.	Only Complete Member of abstract class be static

3 – Why we need equals and hashCode ? When to override ?

We know that two objects are considered equal only if their references point to the same object, and unless we [override equals and hashCode methods](#), the class object will not behave properly on hash-based collections like `HashMap`, `HashSet`, and `Hashtable`. This is because hash-based collections are organized like a sequence of buckets, and the hash code value of an object is used to determine the bucket where the object would be stored, and the same hash code is used again to find the object's position in the bucket. The key retrieval is basically a two-step process:

1. Finding the correct bucket using `hashCode()` method.
2. Linearly searching the bucket for the key using `equals()` method.

4 – Diamond problem in Java ? How to fix it?

In Java, The Diamond Problem Is Related To Multiple Inheritance. Sometimes It Is Also Known As The Deadly Diamond Problem Or Deadly Diamond Of Death. In This Section, We Will Learn What Is The Demand Problem In Java And What Is The Solution To The Diamond Problem.

THE SOLUTION OF DIAMOND PROBLEM

the solution to the diamond problem is default methods and interfaces. we can achieve multiple inheritance by using these two things.

the default method is similar to the abstract method. the only difference is that it is defined inside the interfaces with the default implementation. we need not to override these methods. because they are already implementing these interfaces.

the advantage of interfaces is that it can have the same default methods with the same name and signature in two different interfaces. it allows us to implement these two interfaces, from a class. we must override the default methods explicitly with its interface name.

5 – Why we need Garbage Collector ? How does it run ?

Garbage collection in Java is the process by which Java programs perform automatic memory management. Java programs compile to bytecode that can be run on a Java Virtual Machine, or JVM for short. When Java programs run on the JVM, objects are created on the heap, which is a portion of memory dedicated to the program. Eventually, some objects will no longer be needed. The garbage collector finds these unused objects and deletes them to free up memory.

6 – Java 'static' keyword usage ?

The **static keyword** in Java is mainly used for memory management. The static keyword in Java is used to share the same variable or method of a given class. The users can apply static keywords with variables, methods, blocks, and nested classes. The static keyword belongs to the class than an instance of the class. The static keyword is used for a constant variable or a method that is the same for every instance of a class.

The *static* keyword is a non-access modifier in Java that is applicable for the following:

1. Blocks
2. Variables
3. Methods
4. Classes

7 – Immutability means ? Where, How and Why to use it ?

final : In Java, [final](#) is a modifier which is used for class, method and variable also. When a variable is declared with final keyword, it's value can't be modified, essentially, a constant.

[Immutability](#) : In simple terms, immutability means unchanging over time or unable to be changed. In Java, we know that String objects are immutable means we cant change anything to the existing String objects.

8 – Composition and Aggregation means and differences ?

Association is a relation between two separate classes which establishes through their Objects. Association can be one-to-one, one-to-many, many-to-one, many-to-many. In Object-Oriented programming, an Object communicates to another object to use functionality and services provided by that object. **Composition** and **Aggregation** are the two forms of association.

9 – Cohesion and Coupling means and differences ?

Cohesion:

Cohesion is the indication of the relationship within module. It is concept of intra-module. Cohesion has many types but usually highly cohesion is good for software.

Coupling:

Coupling is also the indication of the relationships between modules. It is concept of Inter-module. Coupling has also many types but usually low coupling is good for software.

Now we will see the difference between Cohesion and Coupling. the differences between cohesion and coupling are given below:

Cohesion

Cohesion is the concept of intra module.

Cohesion represents the relationship within module.

Coupling

Coupling is the concept of inter module.

Coupling represents the relationships between modules.

Increasing in cohesion is good for software.

Increasing in coupling is avoided for software.

Cohesion represents the functional strength of modules.

Coupling represents the independence among modules.

Highly cohesive gives the best software.

Where as loosely coupling gives the best software.

In cohesion, module focuses on the single thing.

In coupling, modules are connected to the other modules.

10 - Heap and Stack means and differences ?

Parameter	STACK	HEAP
Basic	Memory is allocated in a contiguous block.	Memory is allocated in any random order.
Allocation and De-allocation	Automatic by compiler instructions.	Manual by the programmer.
Cost	Less	More
Implementation	Easy	Hard
Access time	Faster	Slower
Main Issue	Shortage of memory	Memory fragmentation
Locality of reference	Excellent	Adequate
Safety	Thread safe, data stored can only be accessed by owner	Not Thread safe, data stored visible to all threads
Flexibility	Fixed-size	Resizing is possible

Parameter	STACK	HEAP
Data type structure	Linear	Hierarchical

11 – Exception means ? Type of Exceptions ?

An exception is an unwanted or unexpected event, which occurs during the execution of a program i.e at run time, that disrupts the normal flow of the program's instructions.

There are two type exception.

Built-in exceptions, User defined exceptions

12 – How to summarize 'clean code' as short as possible ?

Code is clean if it can be understood easily – by everyone on the team.

Clean code can be read and enhanced by a developer other than its original author. With understandability comes readability, changeability, extensibility and maintainability.

13 - What is the method of hiding in Java ?

Method hiding can be defined as, "if a subclass defines a static method with the same signature as a static method in the super class, in such a case, the method in the subclass hides the one in the superclass." The mechanism is known as **method hiding**. It happens because static methods are resolved at compile time.

14 - What is the difference between abstraction and polymorphism in Java ?

Data abstraction means information hiding. Usually what is hidden is the representation of a data structure. Example: I implement sets, but I don't tell you whether a set is represented as a list, a balanced binary tree, or an unbalanced binary tree. Done right, I can change representation without breaking your code.

Polymorphism means reuse with different types. So with my set example you could create sets of Social Security numbers, sets of full names, or sets of fruitbats, all using the same code

Resources:

<https://www.patika.dev> NYP nedir

<https://stackoverflow.com/questions/761194/interface-vs-abstract-class-general-oo> (take photo link website)

<https://www.techiedelight.com/why-override-equals-and-hashcode-methods-java/>

<https://i.stack.imgur.com/Xf8Yz.png> (Photo Link)

<https://www.javatpoint.com/what-is-diamond-problem-in-java>

<https://www.geeksforgeeks.org/garbage-collection-java/>

<https://www.geeksforgeeks.org/static-keyword-java/#:~:text=The%20static%20keyword%20in%20Java,an%20instance%20of%20the%20class.>

<https://www.geeksforgeeks.org/final-vs-immutability-java/>

<https://www.geeksforgeeks.org/software-engineering-differences-between-coupling-and-cohesion/>

<https://www.geeksforgeeks.org/types-of-exception-in-java-with-examples/>

<https://gist.github.com/wojteklu/73c6914cc446146b8b533c0988cf8d29>

<https://www.javatpoint.com/method-hiding-in-java>