

[By OnlineInterviewQuestions.com](http://OnlineInterviewQuestions.com)

[Java Inheritance Interview Questions](#)

Practice Best Java Inheritance Interview Questions and Answers

Java inheritance is a methodology in oops which deals with the familiarization of the object-oriented programming concepts of Java. It forms one of the most integral parts of this programming language in which one object can get the inherited features as well as the data contained in the other classes of the program.

Finally practice here the best Java Inheritance Interview Questions and Answers, for the best preparation of the Java Inheritance Interview. apart from this, you can also download here **Java Inheritance Interview Questions PDF**, completely free.

Q1. [Explain Inheritance in Java?](#)

Inheritance is one of the most important pillars of the OOPs (Object Oriented programming system). Inheritance in Java is a mechanism by which one class acquires all the properties and behaviors of another class. The class from which the subclass is derived is known as the superclass (also a base class or a parent class). The idea behind inheritance in Java is that you can create subclasses that are built upon existing superclasses. When a class is derived from an existing class, you can reuse methods and fields of the superclass in it. Moreover, you can add new methods and fields in the subclass also. Inheritance represents the parent-child relationship which is also known as an IS-A relationship. In other words, You can say that the class from which another class acquires the properties and behavior is known as the superclass (sometimes base class or parent class). The class that acquires the properties and behavior of an existing superclass is known as the subclass (sometimes derived class or child class).

Q2. [Enlist different types of Inheritance supported by Java?](#)

The various kinds of inheritance supported by Java.

- **Single Inheritance:** Single Inheritance is the easy inheritance of all, while a group lengthens another group (Simply one class) then one calls it as Single inheritance.
- **Hierarchical Inheritance:** In Hierarchical inheritance, one mother group will be received by multiple subclasses.
- **Multilevel Inheritance:** In Multilevel Inheritance, a derived group will be inheriting a mother group and as also as the derived group performance as the mother group to other groups.
- **Hybrid Inheritance (Through Interface):** Hybrid Inheritance is the mixture of both Multiple and Single Inheritance.

Q3. What is the difference between Inheritance and Encapsulation?

The major **difference between Inheritance and Encapsulation** are as follows -

Inheritance

Inheritance is the process or mechanism by which you can acquire the properties and behavior of a class into another class.

Inheritance indicates that a child class (subclass) inherits all the attributes and methods from a parent class (superclass).

/* Following is an example that demonstrates how to achieve Inheritance in Java */

```
class Animal{
void eat(){
System.out.println("eating...");
}
}
class Dog extends Animal{ void bark(){
System.out.println("barking...");
}
}
class TestInheritance{
public static void main(String args[]){
Dog d=new Dog();
d.bark(); d.eat();
}}
```

Encapsulation

Encapsulation refers to the winding of data into a single unit which is known as class.

Encapsulation indicates that one class must not have access to the (private) data of another class.

/* Following is an example that demonstrates how to achieve Encapsulation in Java */

```
public class EncapTest {
private String name;
private String idNum;
private int age;
public int getAge() {
return age;
}
public String getName() {
return name;
}
public String getIdNum() {
return idNum;
}
public void setAge( int newAge) {
age = newAge;
}
public void setName(String newName) {
name = newName;
}
public void setIdNum( String newId) {
idNum = newId;
} }
}
```

Q4. Are static members inherited to sub classes?

Yes! static members are inherited to sub classes.

Q5. How to restrict a member of a class from inheriting by its sub classes?

Private members are not used by anything. Therefore, the Programmer can restrict a member of a class from inheriting its subclasses by using a private access specifier.

Q6. Explain covariant method overriding in Java.

Covariant method overriding helps to remove typecasting on the client-side. It allows you to return a subtype of the overridden method.

Q7. Explain aggregation in java?

In Java, **Aggregation** is described as a "has-a" and "whole/part" relationship that shows the relationship between two classes. The main purpose of Aggregation is reusability. For instance, A car object is an aggregation of an engine, seat, wheels, and other objects.

Q8. Why Multiple inheritance is not supported by Java?

Multiple Inheritance is a mechanism by which a subclass acquires the properties of more than one class. A class cannot be derived from more than one class in JAVA. For instance, consider a case where class C extends class A and class B and both class A and B have the same method display() because the java compiler cannot decide which display method it should inherit. In Java, The main reason behind the impossibility of multiple inheritances is to prevent ambiguity. To overcome this problem, The concept of interface is adopted in Java by which a class can implement one or more interfaces to achieve multiple inheritances.

Q9. What is multi level inheritance in Java?

Multilevel inheritance refers to a mechanism by which you can easily acquire the properties and behaviour of a class into some other classes. In multilevel inheritance, a class C can acquire the properties and behaviour of a derived class B that is derived from class A. Now the derived class B becomes the base class and the new class C becomes the derived class. When a class extends a class, which extends another class then this mechanism is called multilevel inheritance. For instance, class C extends class B and class B extends class A then this type of inheritance is known as multilevel inheritance.

Q10. Enlist few advantages of inheritance?

Few advantages of inheritance are listed below:

- Inheritance reduces code redundancy.
- It provides code reusability.

- It reduces source code size and improves code readability.
- It generates a Manageable code.

Q11. What is use of super keyword in Java?

The super keyword is used as precedence with method and constructor to call superclass's methods and to access the superclass's constructor. It also eliminates the confusion between superclasses and subclasses that have methods with the same name.

Q12. Explain Abstract class in Java?

A class that is declared using the “abstract” keyword is known as The abstract class. It can have abstract methods(methods without body) as well as concrete methods (regular methods with the body) whereas a normal class(non-abstract class) cannot have abstract methods. An abstract class can not be instantiated. It does not allow to create an object of it.

Q13. Enlist access specifiers available in Java?

The access specifiers available in JAVA are listed below:

private: The accessibility of a private modifier is only within the class. It cannot be accessible, outside the class.

default: The accessibility of a default modifier is only within the package. It cannot be accessible, outside the package. In the case of the non-specify access level, it will be the default.

protected: The accessibility of a protected modifier is within the package and outside the package through child class. It can be accessible by only child class but cannot be accessible, outside the package

public: The accessibility of a public modifier is everywhere. It can be accessible, within the class, outside the class, within the package, and outside the package.

Please Visit OnlineInterviewquestions.com to download more pdfs