

Interview Questions | MCQs

HOME Interview Questions MCQs Class Notes *LAB VIVA SEMINAR TOPICS
ONLINE TEST GATE CAT Internship ABOUT US

Any Skill Set

Download &
Customize
Chrome

Google

Download
&
Custom
Chrome

Google

[Home](#) » [Java Programming With Oops Concepts Questions](#) » **300+
[LATEST] Java Programming WithOops Concepts
Interview Questions and Answers**

**300+ [LATEST] Java Programming
WithOops Concepts Interview
Questions and Answers**

Download



Download the Secure Browser

Google

Q1. What Is Meant By Dynamic Binding?

Dynamic binding is a binding in which the class association is not made until the object is created at execution time. It is also called as late binding.

Q2. What Is The 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.

Download the Secure Browser

Google



Q3. Can You Use Abstract And Final Both With A Method?

Google

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

Q4. What Is The Difference Between Static Binding And Dynamic Binding?

In case of static binding type of object is determined at compile time whereas in dynamic binding type of object is determined at runtime.

Q5. What Is A Subclass?

Subclass is a class that inherits from one or more classes.

Q6. 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.

Q7. What Is An Object?



Object is an instance of a class. It has state, behaviour and identity. It is also called as an instance of a class.

Q8. What Is Blank Or Uninitialized Final Variable?

A final variable that is not initialized at the time of declaration is known as blank final variable.

If you want to create a variable that is initialized at the time of creating object and once initialized may not be changed, it is useful. For example PAN CARD number of an employee. It can be initialized only in constructor.

**Download the
Secure Browser**

Google

Q9. What Is Multiple Inheritance And Does Java Support?

**Download the
Secure Browser**

Google



If a child class inherits the property from multiple classes is known as multiple inheritance. Java does not allow to extend multiple classes. The problem with multiple inheritance is that if multiple parent classes have a same method name, at runtime it becomes difficult for compiler to decide which method to execute from the child class. To overcome this problem it allows to implement multiple Interfaces.

Q10. What Is Meant By Polymorphism?

Polymorphism literally means taking more than one form. Polymorphism is a characteristic of being able to assign a different behavior or value in a subclass, to something that was declared in a parent class.

Q11. What Is A Destructor?

Destructor is an operation that frees the state of an object and/or destroys the object itself. In Java, there is no concept of destructors. It's taken care by the JVM.

Q12. What Is Aggregation?

Aggregation is a specialized form of Association where all objects have their own lifecycle but there is ownership and child object can not belong to another parent object. Let's take an example of Department and teacher. A single teacher can not belong to multiple departments, but if we delete the department teacher object will not destroy. We can think about "has-a" relationship.

Q13. What Are Pass By Reference And Pass By Value?

When an object is passed by value, this means that a copy of the object is passed. Thus, even if changes are made to that object, it doesn't affect the original value. When an object is passed by reference, this means that the actual object is not passed, rather a reference of the object is passed. Thus, any changes made by the external method, are also reflected in all places.

Q14. 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.

Q15. Difference Between Method Overloading And Overriding.

Method Overloading :

- Method overloading increases the readability of the program.
- method overlaoding is occurs within the class.
- In this case, parameter must be different.

Method Overriding :

- Method overriding provides the specific implementation of the method that is already provided by its super class.
- Method overriding occurs in two classes that have IS-A relationship.
- In this case, parameter must be same.

Q16. What Is Collaboration?

Collaboration is a process whereby several objects cooperate to provide some higher level behavior.

Q17. 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.

Q18. What Will Be The Initial Value Of An Object Reference Which Is Defined As An Instance Variable?



Download the Secure Browser

Google

The object references are all initialized to null in Java.

Q19. What Is Difference Between Polymorphism And Inheritance?

- Inheritance defines parent-child relationship between two classes, polymorphism take advantage of that relationship to add dynamic behaviour in your code.
- Inheritance helps in code reusability by allowing child class to inherit behavior from the parent class. On the other hand Polymorphism allows Child to redefine already defined behaviour inside parent class. Without Polymorphism it's not possible for a Child to execute its own behaviour while represented by a Parent reference variable, but with Polymorphism he can do that.
- Java doesn't allow multiple inheritance of classes, but allows multiple inheritance of Interface, which is actually required to implement Polymorphism. For example a Class can be Runnable, Comparator and Serializable at same time, because all three are interfaces. This makes them to pass around in code e.g. you can pass instance of this class to a method which accepts Serializable, or to Collections.sort() which accepts a Comparator.
- Both Polymorphism and Inheritance allow Object oriented programs to evolve. For example, by using Inheritance you can define new user types in an Authentication System and by using Polymorphism you can take advantage of already written authentication code. Since, Inheritance guarantees minimum



base class behaviour, a method depending upon super class or super interface can still accept object of base class and can authenticate it.

Q20. What Is Meant By Encapsulation?

Encapsulation is the process of compartmentalizing the elements of an abstraction that defines the structure and behavior. Encapsulation helps to separate the contractual interface of an abstraction and implementation.

Q21. What Is An Abstract Class?

Abstract class is a class that has no instances. An abstract class is written with the expectation that its concrete subclasses will add to its structure and behavior, typically by implementing its abstract operations.

Q22. When Super Keyword Is Used?

If the method overrides one of its superclass's methods, overridden method can be invoked through the use of the keyword super. It can be also used to refer to a hidden field.

Q23. Can There Be Any Abstract Method Without Abstract Class?

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

Q24. 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.



Q25. What Is Meant By Static Binding?

Static binding is a binding in which the class association is made during compile time. This is also called as early binding.

Q26. What Is Meant By Inheritance?

Inheritance is a relationship among classes, wherein one class shares the structure or behavior defined in another class. This is called Single Inheritance. If a class shares the structure or behavior from multiple classes, then it is called Multiple Inheritance. Inheritance defines “is-a” hierarchy among classes in which one subclass inherits from one or more generalized superclasses.

Q27. What Is Covariant Return Type?

The covariant return type specifies that the return type may vary in the same direction as the subclass.

Before Java5, it was not possible to override any method by changing the return type. But now, since Java5, it is possible to override method by changing the return type if subclass overrides any method whose return type is Non-Primitive but it changes its return type to subclass type.

Q28. What Is Meant By Object Oriented Programming?

OOP is a method of programming in which programs are organized as cooperative collections of objects. Each object is an instance of a class and each class belongs to a hierarchy.

Q29. What Is Composition?

Composition is again specialize form of Aggregation and we can call this as a “death” relationship. It is a strong type of Aggregation. Child object does not have their lifecycle and if parent object deletes all child object will also be deleted. Let's take again an example of relationship between House and rooms. House can contain multiple rooms there is



no independent life of room and any room can not belongs to two different house if we delete the house room will automatically delete.

Q30. Can We Override Static Method?

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

Q31. What Is Meant By Binding?

Binding denotes association of a name with a class

Q32. 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.

Q33. What Is Association?

Association is a relationship where all objects have their own lifecycle and there is no owner. Let's take an example of Teacher and Student. Multiple students can associate with single teacher and single student can associate with multiple teachers but there is no ownership between the objects and both have their own lifecycle. Both can create and delete independently.

Q34. What Is Meant By Abstraction?

Abstraction defines the essential characteristics of an object that distinguish it from all other kinds of objects. Abstraction provides crisply-defined conceptual boundaries relative to the perspective of the viewer. It's the process of focusing on the essential characteristics



of an object. Abstraction is one of the fundamental elements of the object model.

Q35. 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 non-abstract 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.

Q36. What Is A Superclass?

Superclass is a class from which another class inherits.

Q37. 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.

Q38. 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.

Q39. What Is Meant By Persistence?

Persistence is the property of an object by which its existence transcends space and time.

---->> **Below Are The Related Posts Of Above Questions :::**

----->>[**MOST IMPORTANT**]<<-----

- 1. [250+ TOP MCQs on Concepts of OOPs and Answers](#)**
- 2. [300+ \[LATEST\] Java Abstraction Interview Questions and Answers](#)**
- 3. [250+ TOP MCQs on OOPs Basic Concepts and Answers](#)**
- 4. [300+ \[LATEST\] Java Inheritance Interview Questions and Answers](#)**
- 5. [300+ TOP OOPS Interview Questions and Answers](#)**



- 6. 300+ [LATEST] Java Programmer Interview Questions and Answers**
- 7. 300+ [LATEST] Java Constructor Interview Questions and Answers**
- 8. 200+ TOP OOPS LAB VIVA Questions and Answers Pdf**
- 9. 250+ TOP MCQs on OOPs Concept – 2 and Answers**
- 10. 300+ [LATEST] Ericsson Java Interview Questions and Answers**
- 11. 300+ [LATEST] Aricent Java Interview Questions and Answers**
- 12. 300+ [LATEST] Java Hadoop Developer Interview Questions and Answers**
- 13. 300+ [LATEST] C# Oops Interview Questions and Answers**
- 14. 250+ TOP MCQs on OOPs Features and Answers**
- 15. 300+ [LATEST] Java.lang Interview Questions and Answers**
- 16. 300+ [LATEST] Java Inner Classes Interview Questions and Answers**
- 17. 300+ [LATEST] Java.lang Package Interview Questions and Answers**
- 18. 300+ TOP Sap Object Oriented Abap Interview Questions and Answers**
- 19. 250+ TOP MCQs on OOPs Concept – 3 and Answers**
- 20. 300+ [MOSK ASKED] Infosys Core Java Interview Questions and Answers**

Engineering 2022 , FAQs Interview Questions , Theme by [Engineering](#)|| [Privacy Policy](#)|| [Terms and Conditions](#)|| [ABOUT US](#)|| [Contact US](#)||

Engineering interview questions,Mcqs,Objective Questions,Class Lecture Notes,Seminor topics,Lab Viva Pdf PPT Doc Book free download. Most Asked Technical Basic CIVIL | Mechanical | CSE | EEE | ECE | IT | Chemical | Medical MBBS Jobs Online Quiz Tests for Freshers Experienced .

