Java Interview

# **Oops Concepts:**

## **Encapsulation:**

1. What is Encapsulation in Java? Why is it called Data hiding and Features?
   * Process of binding data and methods together in a single unit.
   * Encapsulations is used to hide the features and functionalities defined in a class.
   * Encapsulation is the one of the concepts of OOPS
   * Private access modifier helps to keep the variables and method accessible only inside the class helps to hide the data is called data hiding
2. What are the advantages of Encapsulation?
   * Hide data and functionality from outside world
   * Provide secured access to class variables.
   * Has more flexibility in adding new features
   * It improves maintainability of application.
   * A class can have total control over what is stored in its fields.
3. How to achieve encapsulation in Java? Give an example.
   * Assigning variables in a class
   * Give access to variables by getters and setter method.
4. How to achieve Data hiding programmatically?
   * By declaring variables as private and hide from the members outside the class.
5. What is Tightly encapsulated class in Java?
   * Each variable declared as private.
6. What is the difference between Abstraction and Encapsulation?

|  |  |
| --- | --- |
| * + **Abstraction** | * + **Encapsulation** |
| * + Solves problem at design level | * + Solves problem at implementation level |
| * + Implemented using Interface or Abstract class | * + Implemented using private and protected access modifiers |
|  |  |

1. What are the getter and setter methods in Java?
   * They are mutators, used to get access and modify private class variables
2. Explain design pattern based on encapsulation in java? (Revisit)
   * Singleton

## **Inheritance**

1. What is Inheritance in Java?
   * Inheritance is a parent child relationship.
   * The common variables and methods can be declared or implemented.
   * Child can inherent the properties of parent by using extends keyword for class and implements for interface
2. What are different types of Inheritance supported by Java?
   * Single inheritance, multi-level inheritance and multiple inheritance.
   * Single Inheritance can be achieved by class and multiple inheritance is for interface.
3. Why multiple Inheritance is not supported by Java?
   * Diamond problem, if a class extends multiple classes and both parent classes have same method name the child does not know which one to inherit. Interface can implement multiple interfaces.
4. Why Inheritance is used by Java Programmers?
   * To attain the features of parent, and to reduce the code duplication. Leverage Polymorphism.
5. How to use Inheritance in Java?
   * Using extends keyword for class and implements for interface.
6. What is the difference between Inheritance and Encapsulation?

|  |  |
| --- | --- |
| * + Inheritance | * + Encapsulation |
| * + It is used to extend the class or interface to enable polymorphism | * + It is used to hide the variables and methods to other other objects |

1. What is the difference between Inheritance and Abstraction?

|  |  |
| --- | --- |
| * + Inheritance | * + Abstraction |
| * + It is used to extend the class or interface to enable polymorphism | * + It is used in design level   + It forces to implement the method. |

1. What is the difference between Polymorphism and Inheritance?

|  |  |
| --- | --- |
| * + Polymorphism | * + Inheritance |
| * + Polymorphism is used to Overload and overwrite the functionality | * + It is used to extend the class or interface to enable polymorphism |

1. What is the difference between Composition and Inheritance in OOP?

|  |  |
| --- | --- |
| * + Composition | * + Inheritance |
| * + It has HAS-A relationship with other object | * + It has IS-A relationship with other objects |

1. Can we override static method in Java?
   * No, the static methods are resolved at compile time but override methods are available at runtime so static methods cannot be overridden.
2. Can we overload a static method in Java?
   * Yes we can since it will have different parameters
3. Can we override a private method in Java?
   * Yes we can override
4. Can a class implement more than one interface in Java?
   * Yes
5. Can a class extends more than one class in Java?
   * No
6. Can an interface extends more than one interface in Java?
   * Yes
7. What will happen if a class extends two interfaces and they both have a method with same name and signature?
   * Class will implement the override method from both interfaces, which is same in this case
8. Can we pass an object of a subclass to a method expecting an object of the super class?
   * Yes.
9. What is the Liskov substitution principle?
   * The Liskov substitution principle is one of the five object-oriented design principles, collectively know as [SOLID principles](http://javarevisited.blogspot.com/2012/03/10-object-oriented-design-principles.html). This design principle is L of SOLID acronym. The Liskov substitution principle states that in an object oriented program if a function or method is expecting an object of base class then  it should work fine with a derived class object as well. If it cannot function properly with derived class object then the derived class is violating the Liskov Substitution principle.
10. How to call a method of a subclass, if you are holding an object of the subclass in a reference variable of type superclass?
    * By casting super class variable with subclass.

## **Ploymorphism**

1. What is polymorphism and what are the types of it?
   * Single task can be implemented in different ways.
   * Method over loading and method overriding
2. What is method overriding?
   * Specific implementation of a method for child class.
3. What is method overloading?
   * If a class have same method name but different parameters.
4. Difference between method overloading and overriding?

|  |  |
| --- | --- |
| * + Overloading | * + Overriding |
| * + Signature of method changes   + Can be done in same class   + Overloaded method is bonded with static binding | * + Signature of method remains same.   + Must be done in subclass   + Override is bonded with dynamic binding. |

1. What is static and dynamic binding?
   * static binding type of object is determined at compile time whereas in dynamic binding type of object is determined at run time.
2. Why method overloading is not possible by changing the return type in java?
   * If we are just changing the return type this will method override and it is possible within same class.
   * Overloaded method can have different return types for different input parameters, not possible with same number of input params
3. can we overload main() method?
   * Yes we can have by overloading.
4. What is run time polymorphism and compile time polymorphism?
   * Overriding is a runtime, Overloading is a compile time