

QUESTIONS

- 1.What is polymorphism in Java?
- 2.Explain the two types of polymorphism in Java.
- 3.Can you override a static method in Java?
- 4.Explain the concept of "upcasting" in Java.
- 5.Explain the difference between compile-time polymorphism and runtime polymorphism in Java.
- 6.What is the significance of the @Override annotation in Java, especially in the context of polymorphism?
- 7.Discuss the importance of the "IS-A" relationship in the context of polymorphism. How does it relate to inheritance?
- 8.How does polymorphism enhance code flexibility and maintainability in Java programs?
- 9.Discuss the concept of functional interfaces and how they contribute to achieving polymorphism in Java through the introduction of lambda expressions.
- 10.How does the introduction of default methods in interfaces impact polymorphism in Java? Provide an example.
- 11.Explain the limitations and challenges associated with achieving polymorphism in a multithreaded environment in Java.
- 12.Explain the concept of "adapters" and how they can be used to achieve polymorphism in Java. Provide an example where adapters enhance code flexibility.
- 13.What is encapsulation in Java?
- 14.Explain the benefits of encapsulation in Java.
- 15.Why is it important to make variables private in a class?
- 16.What are getter and setter methods, and how do they relate to encapsulation?
- 17.Explain the concept of data hiding in encapsulation.
- 18.Can encapsulation be achieved without using private access modifiers?
- 19.Explain the concept of an immutable class. How does it relate to encapsulation?
- 20.Discuss the challenges and potential issues associated with overusing encapsulation.
- 21.How does encapsulation relate to the principle of information hiding?
- 22.Explain the difference between encapsulation and abstraction.
- 23.How does encapsulation contribute to the principle of "loose coupling" in software design?
- 24.Explain the difference between encapsulation and encapsulation in Java.
- 25.How does encapsulation contribute to the concept of "state encapsulation" in Java classes?
- 26.Discuss the differences between encapsulation and data hiding.
- 27.Why is it recommended to use getter and setter methods instead of making variables public in a class?
- 28.Explain the concept of "package-private" access in Java and how it relates to data hiding within a package.
- 29.What are different ways to achieve or implement polymorphism in Java?
- 30.What are the differences between Polymorphism and Inheritance in Java?
- 31.How Java compiler differentiate between methods in Compile time Polymorphism?
- 32.How to achieve/implement dynamic polymorphism in Java?
- 33.What is Binding in Java?
34. What is Static binding in Java?
- 35.How Java compiler performs static binding?
- 36.What is the dynamic binding?
- 37.What is the difference between late and early binding?
- 38.How JVM performs dynamic binding in Java?
- 39.Is it possible to achieve polymorphism using data members in Java?
- 40.What is constructor overloading?
- 41.What happens if two methods have different return types but share the same name and signature?
- 42.How does the Java Virtual Machine (JVM) handle encapsulation at runtime?

- 43.Explain the concept of "heap memory" and how encapsulation influences the allocation and deallocation of objects in the heap.
- 44.Can you give some examples of when runtime polymorphism may not work as expected?
- 45.Explain the memory implications of method overriding and dynamic method dispatch in polymorphic scenarios.