Java Inheritance - Q&A

1. Can I access parent class properties and behaviours using parent class reference?

Yes, you can access all public and protected members of the parent class using its reference. Private members are not accessible.

2. Can I access parent class properties and behaviours using child class reference?

Yes, you can. A child class inherits public and protected members of the parent class. Private members are still not accessible.

3. Can I access parent class private properties and behaviours using parent class reference from child class?

No, private members of a parent class are not accessible outside the parent class, even in the child class.

4. Can I access child class properties and behaviours using child class reference?

Yes, you can access both parent-inherited members and child-specific members using a child class reference.

5. Can I access parent class properties and behaviours using child class reference?

Yes, you can access inherited public and protected parent members using a child class reference.

6. Can I store child class object into parent class reference variable?

Yes, this is called Upcasting. A parent class reference can hold a child class object, but only parent class members are accessible.

7. Which keyword is used to make relation between classes?

The 'extends' keyword is used to inherit from a class. For interfaces, 'implements' is used.

8. How many types of inheritance are there in Java?

1) Single Inheritance

2) Multilevel Inheritance

3) Hierarchical Inheritance

(Note: Multiple inheritance with classes is not supported, but can be achieved via interfaces.)

9. Can I access child class properties and behaviours using parent class reference?

No, you cannot directly access child-specific members using a parent class reference. However, if the object is actually a child type, you can downcast to access them.

10. When does java.lang.ClassCastException occur?

It is raised when you try to cast an object to a subclass of which it is not an instance. Example: Parent p = new Parent(); Child c = (Child)p; // throws ClassCastException.