**1. In what ways are objects in JavaScript different from objects in Java?**

In **JavaScript**, objects are super flexible! You can add, change, or remove things from an object whenever you want. They don’t have strict rules about what they should look like.

In **Java**, objects are more like boxes that need to be perfectly planned ahead. You need to tell the computer exactly what should be in the object and follow that plan, no changing allowed after it's made.

**2. How do we define "members" of an object in JavaScript, compared to Java?**

In **JavaScript**, "members" (like variables or functions) are just added directly to the object, almost like sticking labels on it. You can just write object.property = value to add something.

In **Java**, members need to be declared inside the object when it’s being made. You have to write things like int age; inside the object to tell Java what’s inside the box.

**3. "Objects are passed by reference." Explain what this means.**

Imagine you have a toy robot. When you hand the robot to a friend, you're giving them **the robot itself**, not just a picture of it. If your friend paints the robot, you’ll see the changes too, because it’s the same toy robot.

This is how **objects** work when passed by reference. If you give your friend an object (like the robot), they can change it, and you’ll see the changes, because you're both working on the same thing.

**4. Why are the standard set of accessor and mutator methods you would code in a Java class not generally necessary in a JavaScript object?**

In **JavaScript**, you don’t need special "getters" and "setters" (like you do in Java) to access or change parts of an object. You can just directly change the properties anytime.

In **Java**, things are more strict, so you use these special methods to safely control how to get or set values, like asking permission before touching the toy.