

## Common Mistakes for Assignment 2: Polymorphism in Java

Common Mistakes	Hints and Solutions	References Link
Inconsistent Naming Conventions.	Make sure to follow consistent naming conventions for methods and variables	<a href="#">Naming Conventions in Java - GeeksforGeeks</a>
Ignoring the order of Parameters	The order of parameters matters when determining overload	<a href="#">Polymorphism in Java - GeeksforGeeks</a>
Forgetting <code>extends</code> Keyword	Make sure to include the <code>extends</code> keyword followed by the superclass name when defining the subclass.	<a href="#">Java Inheritance Tutorial - GeeksforGeeks</a>
Omitting <code>@Override</code> Annotation	Always use <code>@Override</code> before the method in the subclass to indicate that you are intentionally overriding a method in the superclass.	<a href="#">Method Overriding in Java - Oracle Docs</a>
<b>Creating methods Using Final or Static</b>	Methods declared as final or static cannot be overridden, which is a fundamental requirement for polymorphism	<a href="#">Method Overriding in Java - Oracle Docs</a>