Explanation:

Static methods belong to the class, not an instance:

Static methods are associated with the class itself, not with specific objects created from that class. They are resolved at compile time based on the class type, not at runtime based on the object's actual type.

Overriding relies on dynamic method dispatch:

Overriding is a concept of polymorphism that relies on dynamic method dispatch, where the specific method implementation to be executed is determined at runtime based on the actual object type.

Method Hiding instead of Overriding:

If a subclass defines a static method with the same name and signature as a static method in its superclass, it is not considered overriding. Instead, it is known as "method hiding." The static method in the subclass "hides" the static method in the superclass, meaning that when you call the static method using the subclass's name, you invoke the subclass's version, but the superclass's version remains accessible by explicitly referencing the superclass.

In essence, because static methods are tied to the class and resolved at compile time, they cannot participate in the dynamic polymorphism mechanism that enables method overriding.