## Solution Review: Override a Method using the Super Keyword

This review provides a detailed analysis to solve the 'Override a Method using the Super Keyword' challenge.

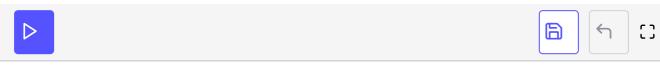
## WE'LL COVER THE FOLLOWING ^

- Solution
  - Explanation

## Solution #

```
// Base Class
class Shape {
  // Private Data Members
  private String name;
  public Shape() { // Default Constructor
        name = "Shape";
  // Getter Function
  public String getName() {
    return name;
}
// Derived Class
class XShape extends Shape {
  private String name;
  public XShape(String name) { // Default Constructor
    this.name = name;
  // Overridden Method
  public String getName() {
    return super.getName() + ", " + this.name;
}
```

```
class Demo {
  public static void main(String args[]) {
    Shape circle = new XShape("Circle");
    System.out.println(circle.getName());
  }
}
```



## **Explanation** #

• Line 29: The super() keyword is used to call the Base Class method
getName(). With the help of the super keyword, the method returns the
base class name, and then we append it with the derived class name
before returning.