Solution Review 1: Override a Method using the Super Function

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

WE'LL COVER THE FOLLOWING

Solution

```
# Parent Class
class Shape:
    sname = "Shape"

    def getName(self):
        return self.sname


# child class
class XShape(Shape):
    # initializer
    def __init__(self, name):
        self.xsname = name

    def getName(self):  # overriden method
        return (super().getName() + ", " + self.xsname)


circle = XShape("Circle")
print(circle.getName())
```

Explanation

- **Line 15:** The `super()` function is used to call the **parent class** method `getName()`. With the help of the `super()` function, the `getName()` method returns the parent class `sname`, and which then gets appended with the

derived class `xsname` before returning.