Challenge 1: Override a Method Using the Super Function

In this challenge, you will override a method using super().

Problem Statement

When a method in a derived class overrides a method in a base class, it is still possible to call the overridden method using the `super()` function.

> If you write `super().method()`, it will call the method that was defined in the superclass.

You are given a partially completed code in the editor. Modify the code so that the code returns the following:

Sample Input

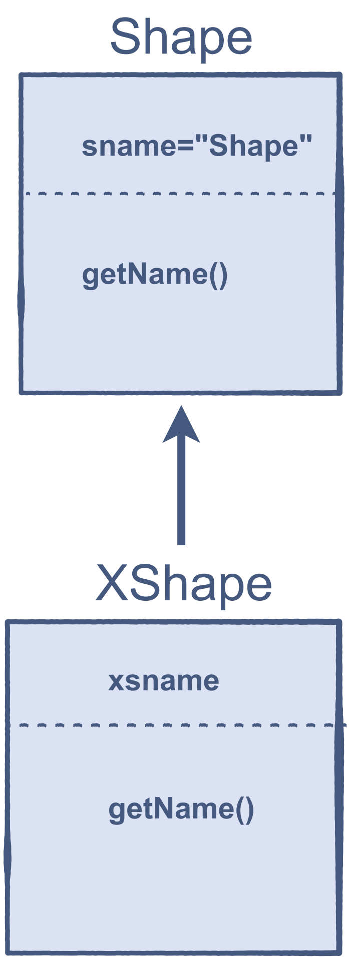
```
circle = XShape("Circle");
circle.getName()
```

Sample Output

```
"Shape, Circle"
```

The `Shape` class is already prepended in the code and it has one property:

sname and one method: getName(). getName() returns sname.



Parent and child class structures

Show Hint

Coding Exercise

First, take a close look and design a step-by-step algorithm before trying the implementation. This problem is designed for your practice, so initially try to solve it on your own. If you get stuck, you can always refer to the solution provided in the solution review.

Good luck!

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

    def getName(self):  # overriden method
        return (self.xsname)
```

The solution will be explained in the next lesson.