In [1]:

```
#Product Pricing System (Property Decorators)
#You are developing a product pricing system. Implement a Product class that:
#● Has an attribute price that is only allowed to be positive.
#● Uses a property decorator for the price to ensure the price is never negative
#● Provide getter, setter, and deleter for the price attribute.
#Task:
#1. Implement the Product class with the necessary decorators.
#2. Create a product with an initial price and try to set a negative price to se
#3. Test deleting the price and observe the system’s behavior when accessing the

class Product:
    def __init__(self, price):
        self._price = None
        self.price = price

    @property
    def price(self):
        """Getter for price"""
        return self._price

    @price.setter
    def price(self, value):
        """Setter for price – ensures only positive values"""
        if value < 0:
            raise ValueError("Price cannot be negative!")
        self._price = value

    @price.deleter
    def price(self):
        """Deleter for price – sets it to None"""
        print("Deleting price...")
        self._price = None

product = Product(100)
print("Initial Price:", product.price)

try:
    product.price = -50
except ValueError as e:
    print("Error:", e)

del product.price
print("Price after deletion:", product.price)
```

```
Initial Price: 100
Error: Price cannot be negative!
Deleting price...
Price after deletion: None
```

In [ ]: