

In [1]: *#Custom String Class: Implement a CustomString class that overloads the + operator for concatenation and the \* operator to repeat the string. Include a method to return the string in uppercase.*

```
class CustomString:
    def __init__(self, value):
        if not isinstance(value, str):
            raise TypeError("CustomString can only be initialized with a string.")
        self.value = value

    def __add__(self, other):
        if not isinstance(other, CustomString):
            raise TypeError("Only CustomString instances can be concatenated.")
        return CustomString(self.value + other.value)

    def __mul__(self, times):
        if not isinstance(times, int):
            raise TypeError("The multiplier must be an integer.")
        return CustomString(self.value * times)

    def to_uppercase(self):
        return self.value.upper()

    def __str__(self):
        return self.value

s1 = CustomString("Hello")
s2 = CustomString("World")

s3 = s1 + s2
print("Concatenated:", s3)

s4 = s1 * 3
print("Repeated:", s4)

print("Uppercase:", s1.to_uppercase())
```

Concatenated: HelloWorld  
Repeated: HelloHelloHello  
Uppercase: HELLO

In [ ]: