In [3]:

```
class Matrix:
    def __init__(self, data):
        self.data = data
        self.rows = len(data)
        self.cols = len(data[0])

    def __add__(self, other):
        if self.rows != other.rows or self.cols != other.cols:
            raise ValueError("Matrices must have the same dimensions to add.")

        result = [
            [self.data[i][j] + other.data[i][j] for j in range(self.cols)]
            for i in range(self.rows)
        ]
        return Matrix(result)

    def __str__(self):
        return '\n'.join([' '.join(map(str, row)) for row in self.data])


# Example
m1 = Matrix([[1, 2], [3, 4]])
m2 = Matrix([[5, 6], [7, 8]])
print("Matrix Sum:\n", m1 + m2)
```

```
Matrix Sum:
 6 8
10 12
```

In [5]:

```
class CustomString:
    def __init__(self, text):
        self.text = text

    def __add__(self, other):
        return CustomString(self.text + other.text)

    def __mul__(self, times):
        return CustomString(self.text * times)

    def to_upper(self):
        return self.text.upper()

    def __str__(self):
        return self.text


# Example
s1 = CustomString("Hello")
s2 = CustomString("World")
print("Concatenation:", s1 + s2)
print("Repeated:", s1 * 3)
print("Uppercase:", s1.to_upper())
```

```
Concatenation: HelloWorld
Repeated: HelloHelloHello
Uppercase: HELLO
```

In [7]:

```
class Currency:
    rates = {
        'USD': 1.0,
        'EUR': 0.85,
        'INR': 75.0
    }

    def __init__(self, amount, currency):
        self.amount = amount
        self.currency = currency

    def convert_to(self, new_currency):
        if new_currency not in Currency.rates:
            raise ValueError("Unsupported currency")

        usd_amount = self.amount / Currency.rates[self.currency]
        converted_amount = usd_amount * Currency.rates[new_currency]
        return Currency(converted_amount, new_currency)

    def __add__(self, other):
        if self.currency != other.currency:
            raise ValueError("Currencies must match to add")
        return Currency(self.amount + other.amount, self.currency)

    def __str__(self):
        return f"{self.amount:.2f} {self.currency}"


# Example
c1 = Currency(100, "USD")
c2 = Currency(50, "USD")
print("Total:", c1 + c2)
converted = c1.convert_to("INR")
print("Converted:", converted)
```

```
Total: 150.00 USD
Converted: 7500.00 INR
```

In [9]:

```
class Rectangle:
    def __init__(self, width, height):
        self.width = width
        self.height = height

    def __lt__(self, other):
        return self.width < other.width

    def __eq__(self, other):
        return self.width == other.width

    def __gt__(self, other):
        return self.width > other.width

    def area(self):
        return self.width * self.height

    def __str__(self):
        return f"Rectangle({self.width}, {self.height})"


# Example
r1 = Rectangle(4, 5)
r2 = Rectangle(6, 3)
```

```
print(r1 < r2)   # True
print(r1 == r2)  # False
print(r1 > r2)   # False
```

```
True
False
False
```

In [ ]: