In [1]:
```
#Currency Conversion: Create a Currency class that represents a monetary amount
#in a specific currency. Overload the + operator to add amounts in the same curr
#and implement a method to convert between different currencies.

class Currency:
    exchange_rates = {
        'USD': 1.0,
        'EUR': 0.85,
        'INR': 83.0,
        'JPY': 110.0,
        'GBP': 0.75,
    }

    def __init__(self, amount, currency):
        if currency not in Currency.exchange_rates:
            raise ValueError(f"Unsupported currency: {currency}")
        self.amount = float(amount)
        self.currency = currency

    def __add__(self, other):
        if not isinstance(other, Currency):
            raise TypeError("Only Currency instances can be added.")
        if self.currency != other.currency:
            raise ValueError("Cannot add amounts in different currencies.")
        return Currency(self.amount + other.amount, self.currency)

    def convert_to(self, target_currency):
        if target_currency not in Currency.exchange_rates:
            raise ValueError(f"Unsupported target currency: {target_currency}")

        amount_in_usd = self.amount / Currency.exchange_rates[self.currency]
        converted_amount = amount_in_usd * Currency.exchange_rates[target_curren
        return Currency(converted_amount, target_currency)

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

usd = Currency(100, "USD")
eur = Currency(85, "EUR")
inr = Currency(1000, "INR")

usd2 = Currency(50, "USD")
print("Addition:", usd + usd2)

converted = usd.convert_to("INR")
print("Converted to INR:", converted)
```

```
Addition: 150.00 USD
Converted to INR: 8300.00 INR
```

In [ ]:
```

```