In [1]:
```python
class Product:
    total_products = 0

    def __init__(self, name, price, stock):
        self.name = name
        self.price = price
        self.stock = stock
        Product.total_products += 1

    def update_stock(self, quantity):
        self.stock += quantity

    def display_info(self, format_type="basic"):
        if format_type == "basic":
            return f"Product: {self.name} | Price: ₹{self.price}"
        elif format_type == "detailed":
            return f"Product: {self.name} | Price: ₹{self.price} | Stock: {self.
        else:
            return "Invalid format type"

    @staticmethod
    def product_info():
        return "Products have a name, price, and stock quantity."

    @classmethod
    def get_total_products(cls):
        return cls.total_products

class Customer:
    customer_count = 0

    def __init__(self, name, email):
        self.name = name
        self.email = email
        self.order_history = []
        Customer.customer_count += 1

    def place_order(self, order):
        if isinstance(order, Order):
            self.order_history.append(order)

    @staticmethod
    def customer_info():
        return "Customers have a name, email, and can place orders."

    @classmethod
    def get_customer_count(cls):
        return cls.customer_count

class Order:
    order_count = 0

    def __init__(self, order_id, customer):
        self.order_id = order_id
        self.customer = customer
        self.products = {}
        Order.order_count += 1
```

```
    def add_product(self, product, quantity):
        if product.stock >= quantity:
            self.products[product] = quantity
            product.update_stock(-quantity)
        else:
            print(f"Not enough stock for {product.name}.")

    @staticmethod
    def order_info():
        return "Orders have a unique ID, a customer, and a list of products with

    @classmethod
    def get_order_count(cls):
        return cls.order_count

p1 = Product("Laptop", 50000, 10)
p2 = Product("Mouse", 500, 50)

c1 = Customer("Alice", "alice@example.com")

o1 = Order("ORD001", c1)
o1.add_product(p1, 2)  # Deducts stock
o1.add_product(p2, 3)

c1.place_order(o1)

print(p1.display_info("basic"))
print(p1.display_info("detailed"))

print(Product.product_info())
print(Customer.customer_info())
print(Order.order_info())

print("Total products:", Product.get_total_products())
print("Total customers:", Customer.get_customer_count())
print("Total orders:", Order.get_order_count())

print("\nOrder History for", c1.name)
for order in c1.order_history:
    print(f"Order ID: {order.order_id}")
    for product, qty in order.products.items():
        print(f"  - {product.name}: {qty} units")
```

```
Product: Laptop | Price: ₹50000
Product: Laptop | Price: ₹50000 | Stock: 8
Products have a name, price, and stock quantity.
Customers have a name, email, and can place orders.
Orders have a unique ID, a customer, and a list of products with quantities.
Total products: 2
Total customers: 1
Total orders: 1

Order History for Alice
Order ID: ORD001
  - Laptop: 2 units
  - Mouse: 3 units
```

In [ ]: