# Define the classes as before

class Product:

def \_\_init\_\_(self, name, price, quantity):

self.name = name

self.price = price

self.quantity = quantity

class Inventory:

def \_\_init\_\_(self):

self.products = []

def add\_product(self, product):

self.products.append(product)

def update\_product\_quantity(self, product\_name, quantity):

for product in self.products:

if product.name == product\_name:

product.quantity += quantity

break

def sell\_product(self, product\_name, quantity):

for product in self.products:

if product.name == product\_name:

if product.quantity >= quantity:

product.quantity -= quantity

print(f"{quantity} {product\_name}(s) sold.")

print(f"Remaining quantity of {product\_name}: {product.quantity}")

else:

print("Insufficient stock.")

break

else:

print("Product not found.")

def list\_products(self):

for product in self.products:

print(f"Name: {product.name}, Price: {product.price}, Quantity: {product.quantity}")

# Example usage with user input

inventory = Inventory()

# Add products to inventory using user input

while True:

name = input("Enter product name (or type 'done' to finish adding products): ")

if name.lower() == 'done':

break

price = float(input("Enter price: "))

quantity = int(input("Enter quantity: "))

inventory.add\_product(Product(name, price, quantity))

# List products

print("\nCurrent inventory:")

inventory.list\_products()

# Sell products using user input

while True:

name = input("Enter product name to sell (or type 'done' to finish selling): ")

if name.lower() == 'done':

break

quantity = int(input("Enter quantity to sell: "))

inventory.sell\_product(name, quantity)

# List updated inventory

print("\nUpdated inventory:")

inventory.list\_products()