**1. Economic Order Quantity (EOQ) Calculation**

import math

def eoq(demand, ordering\_cost, holding\_cost):

"""

Calculate Economic Order Quantity (EOQ)

EOQ = sqrt((2 \* demand \* ordering cost) / holding cost)

"""

return math.sqrt((2 \* demand \* ordering\_cost) / holding\_cost)

# Example Usage

demand = 1000 # annual demand

ordering\_cost = 50 # cost per order

holding\_cost = 2 # holding cost per unit per year

print("EOQ:", eoq(demand, ordering\_cost, holding\_cost))

**2. Reorder Point Calculation**

def reorder\_point(demand\_rate, lead\_time, safety\_stock):

"""

Calculate Reorder Point (ROP)

ROP = (demand rate \* lead time) + safety stock

"""

return (demand\_rate \* lead\_time) + safety\_stock

# Example Usage

demand\_rate = 100 # units per day

lead\_time = 5 # days

safety\_stock = 50 # units

print("Reorder Point:", reorder\_point(demand\_rate, lead\_time, safety\_stock))

**3. ABC Analysis for Inventory Classification**

def abc\_analysis(items):

"""

Classify inventory items into A, B, C categories based on value.

"""

total\_value = sum(item["value"] for item in items)

sorted\_items = sorted(items, key=lambda x: x["value"], reverse=True)

a\_threshold = 0.7 \* total\_value

b\_threshold = 0.9 \* total\_value

a\_value = b\_value = 0

for item in sorted\_items:

if a\_value < a\_threshold:

item["category"] = "A"

a\_value += item["value"]

elif b\_value < b\_threshold:

item["category"] = "B"

b\_value += item["value"]

else:

item["category"] = "C"

return sorted\_items

# Example Usage

items = [{"name": "Item1", "value": 5000}, {"name": "Item2", "value": 3000}, {"name": "Item3", "value": 1000}]

classified\_items = abc\_analysis(items)

for item in classified\_items:

print(item)

**4. Simple Demand Forecasting (Moving Average)**

def moving\_average(demand\_history, period):

"""

Calculate Moving Average Forecast

"""

if len(demand\_history) < period:

return sum(demand\_history) / len(demand\_history)

return sum(demand\_history[-period:]) / period

# Example Usage

demand\_history = [100, 120, 130, 140, 150, 160]

period = 3

print("Forecasted Demand:", moving\_average(demand\_history, period))

**5. Safety Stock Calculation**

def safety\_stock(std\_dev\_demand, lead\_time, service\_factor=1.65):

"""

Calculate Safety Stock

Safety Stock = service\_factor \* std\_dev\_demand \* sqrt(lead\_time)

"""

return service\_factor \* std\_dev\_demand \* (lead\_time \*\* 0.5)

# Example Usage

std\_dev\_demand = 20

lead\_time = 5 # days

print("Safety Stock:", safety\_stock(std\_dev\_demand, lead\_time))

**6. Supplier Rating System**

def supplier\_rating(quality, delivery\_time, cost):

"""

Rate a supplier based on quality, delivery, and cost.

Score = (0.5 \* quality) + (0.3 \* delivery\_time) + (0.2 \* cost)

Higher score means better supplier.

"""

return (0.5 \* quality) + (0.3 \* delivery\_time) + (0.2 \* cost)

# Example Usage

supplier1 = supplier\_rating(90, 80, 70)

supplier2 = supplier\_rating(85, 85, 75)

print("Supplier 1 Score:", supplier1)

print("Supplier 2 Score:", supplier2)

**7. Transportation Cost Calculation**

def transport\_cost(distance, weight, rate\_per\_km):

"""

Calculate transportation cost.

Cost = Distance \* Weight \* Rate per Km

"""

return distance \* weight \* rate\_per\_km

# Example Usage

distance = 500 # km

weight = 1000 # kg

rate\_per\_km = 2 # cost per km per kg

print("Transportation Cost:", transport\_cost(distance, weight, rate\_per\_km))

**8. Warehouse Space Utilization**

def warehouse\_utilization(used\_space, total\_space):

"""

Calculate Warehouse Space Utilization Percentage

"""

return (used\_space / total\_space) \* 100

# Example Usage

used\_space = 800 # cubic meters

total\_space = 1000 # cubic meters

print("Warehouse Utilization:", warehouse\_utilization(used\_space, total\_space), "%")

**9. Lead Time Variability Calculation**

def lead\_time\_variability(lead\_times):

"""

Calculate standard deviation of lead times.

"""

mean\_lt = sum(lead\_times) / len(lead\_times)

variance = sum((lt - mean\_lt) \*\* 2 for lt in lead\_times) / len(lead\_times)

return variance \*\* 0.5

# Example Usage

lead\_times = [5, 6, 4, 5, 7, 5, 6]

print("Lead Time Variability:", lead\_time\_variability(lead\_times))

**10. Order Fill Rate Calculation**

def order\_fill\_rate(orders\_fulfilled, total\_orders):

"""

Calculate Order Fill Rate

"""

return (orders\_fulfilled / total\_orders) \* 100

# Example Usage

orders\_fulfilled = 950

total\_orders = 1000

print("Order Fill Rate:", order\_fill\_rate(orders\_fulfilled, total\_orders), "%")