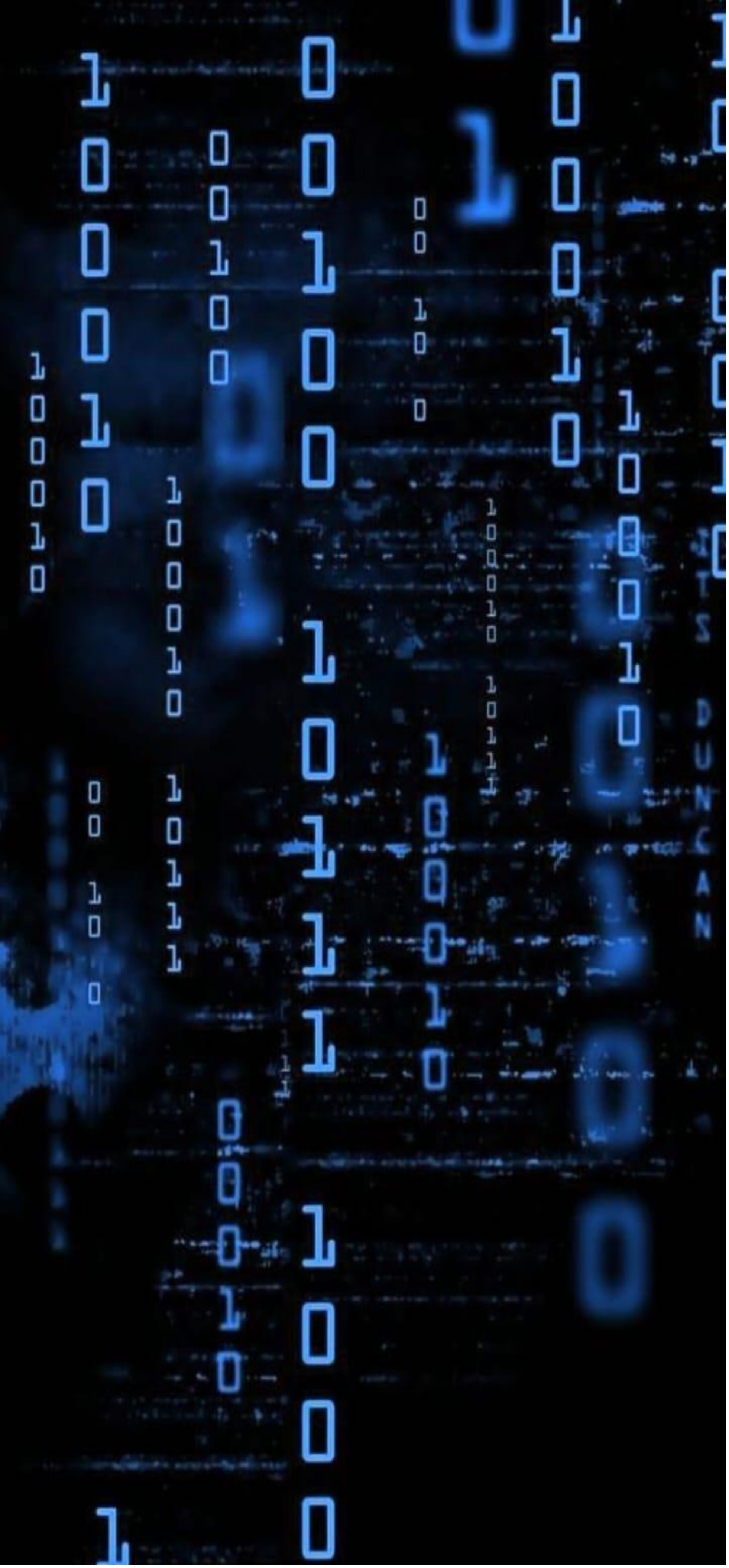


Inventory

System Problem

Index

Problem Definition

We are simulating the inventory system of a company that sells products and needs to manage its stock efficiently.

The goal is to develop an ordering policy (order quantity and reorder point) that minimizes the total inventory-related costs, including holding, ordering, and storage costs.

The simulation will track daily inventory changes, customer demand, and costs, helping to identify an effective inventory management strategy.

Assumptions

- **Initial inventory:** The company starts with 5 units in stock.
- **Reorder Point:** When inventory falls to 3 units or fewer, a new order is placed.
- **Order Quantity:** A fixed quantity is ordered whenever a reorder occurs (specific quantity assumed during simulation).
- **Customer Demand:** Demand varies daily and is randomly generated within a reasonable range.
- **Costs:**
  - Holding cost: $0.10 per unit per day. ○ Ordering cost: $20 per

    order.
  - Storage cost: $50 per unit (may be considered separately if needed).
- **Lead Time:** No delay between placing and receiving an order (inventory replenishes immediately).
- **Shortages:** If demand exceeds available inventory, all available stock is sold (no backorders).
- **Simulation Period:** The system is simulated over a defined number of days (e.g., 5–10 days).

Model Formulation

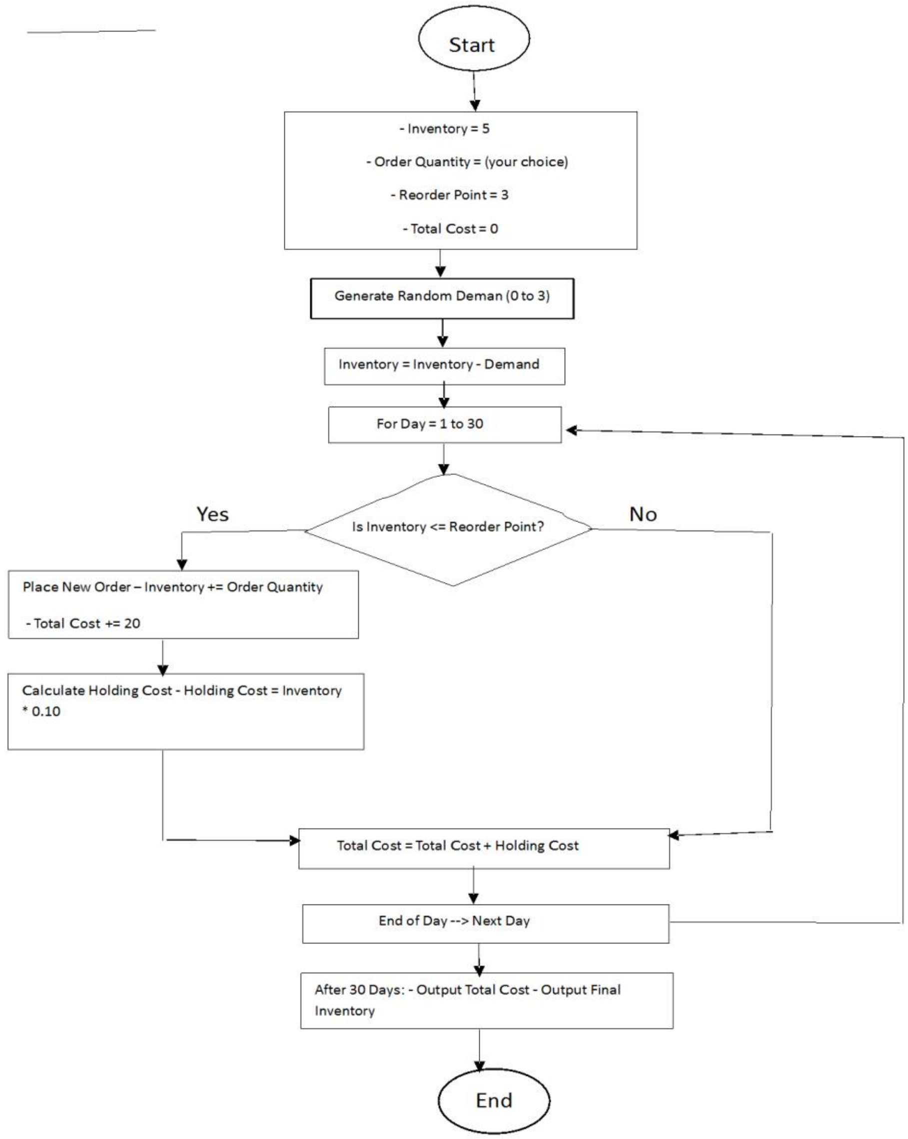
The inventory system is modeled through daily operations as follows:

- **Starting Inventory:** Recorded at the beginning of each day.
- **Customer Demand:** Randomly determined at the start of each day.
- **Ending Inventory:**
  - Calculated as:

  Ending Inventory = max(Starting Inventory - Demand, 0)
- **Reordering Decision:**
  - If Ending Inventory ≤ Reorder Point, an order for the fixed Order Quantity is placed.
- **Cost Calculations:**
  - Holding Cost = Ending Inventory × 0.10 ◦ Ordering Cost = 20 if an order is placed; 0 otherwise ◦ Storage Cost = Ending Inventory × 50 (optional based on case requirement)
  - Total Daily Cost = Holding Cost + Ordering Cost
- **Inventory Update:**
  - If an order was placed, Ending Inventory is updated: Ending Inventory += Order Quantity

The model repeats these steps for each day of the simulation

Flowchart


Start
- Inventory = 5
- Order Quantity = (your choice)
- Reorder Point = 3
- Total Cost = 0
Generate Random Deman (0 to 3)
Inventory = Inventory - Demand
For Day = 1 to 30
Yes
Is Inventory <= Reorder Point?
No
Place New Order – Inventory += Order Quantity
- Total Cost += 20
Calculate Holding Cost - Holding Cost = Inventory * 0.10
Total Cost = Total Cost + Holding Cost
End of Day --> Next Day
After 30 Days: - Output Total Cost - Output Final Inventory
End

Code

```cpp
#include <iostream>
#include <cstdlib>
#include <ctime>

using namespace std;

// Constants
const double holdingCostPerUnit = 0.10;
const double orderingCost = 20.0;
const double storageCostPerUnit = 50.0;
const int orderQuantity = 5;
const int reorderPoint = 3;
const int simulationDays = 30;

// Function to generate random daily demand (between 0 and 3 units)
int generateDemand() {
    return rand() % 4; // 0, 1, 2, or 3
}

int main() {
    srand(time(0));

    int inventory = 5;
    double totalHoldingCost = 0;
    double totalOrderingCost = 0;
    double totalStorageCost = 0;
    int totalOrders = 0;
    int shortageDays = 0;
```

```cpp
    for (int day = 1; day <= simulationDays; day++) {
        int dailyDemand = generateDemand();

        if (inventory < dailyDemand) {
            shortageDays++;
        }

        inventory -= dailyDemand;
        if (inventory < reorderPoint) {
            inventory += orderQuantity;
            totalOrders++;
        }

        // Holding cost = inventory * holding cost per unit
        totalHoldingCost += inventory * holdingCostPerUnit;

        // Storage cost = inventory * storage cost per unit (if inventory > 0)
        if (inventory > 0) {
            totalStorageCost += inventory * storageCostPerUnit;
        }

        cout << day << "," << inventory << "," << dailyDemand << ","
            << (inventory <= reorderPoint ? "Yes" : "No") << "," << inventory << endl;
    }

    totalOrderingCost = totalOrders * orderingCost;

    double totalCost = totalHoldingCost + totalOrderingCost + totalStorageCost;

    cout << "\nSimulation Results:" << endl;
    cout << "Total Holding Cost: $" << totalHoldingCost << endl;
    cout << "Total Ordering Cost: $" << totalOrderingCost << endl;
    cout << "Total Storage Cost: $" << totalStorageCost << endl;
```

```cpp
    cout << "Total Cost: $" << totalCost << endl;
    cout << "Shortage Days: " << shortageDays << endl;

    return 0;
}
```

Screenshot of the output

Microsoft Visual Studio Debug Console

```
Day,Inventory,Daily Demand,Order Placed,Inventory After
1,3,2,Yes,3
2,7,1,No,7
3,4,3,No,4
4,4,0,No,4
5,6,3,No,6
6,4,2,No,4
7,7,2,No,7
8,5,2,No,5
9,3,2,Yes,3
10,3,0,Yes,3
11,6,2,No,6
12,4,2,No,4
13,7,2,No,7
14,4,3,No,4
15,6,3,No,6
16,6,0,No,6
17,4,2,No,4
18,3,1,Yes,3
19,6,2,No,6
20,3,3,Yes,3
21,6,2,No,6
22,5,1,No,5
23,7,3,No,7
24,6,1,No,6
25,4,2,No,4
26,3,1,Yes,3
27,3,0,Yes,3
28,6,2,No,6
29,4,2,No,4
30,7,2,No,7

Simulation Results:
Total Holding Cost: $14.6
Total Ordering Cost: $220
Total Storage Cost: $7300
Total Cost: $7534.6
Shortage Days: 0
```

Statistical Analysis

Day	Beginning Inventory	Daily Demand	Ordered?	Order Quantity	Ending Inventory	Storage Cost	Ordering Cost	Holding Cost	Total Daily Cost
1	5	3	No	0	2	100	0	0.2	2.2
2	2	1	Yes	5	6	300	20	0.6	26.6
3	6	0	No	0	6	300	0	0.6	6.6
4	6	3	No	0	3	150	0	0.3	3.3
5	3	2	Yes	5	6	300	20	0.6	26.6
6	6	3	No	0	3	150	0	0.3	3.3
7	3	2	Yes	5	6	300	20	0.6	26.6
8	6	1	No	0	5	250	0	0.5	5.5
9	5	1	No	0	4	200	0	0.4	4.4
10	4	2	No	0	2	100	0	0.2	2.2
11	2	3	Yes	5	4	200	20	0.4	24.4
12	4	0	No	0	4	200	0	0.4	4.4
13	4	2	No	0	2	100	0	0.2	2.2
14	2	1	Yes	5	6	300	20	0.6	26.6
15	6	1	No	0	5	250	0	0.5	5.5
16	5	1	No	0	4	200	0	0.4	4.4
17	4	3	No	0	1	50	0	0.1	1.1
18	1	1	Yes	5	5	250	20	0.5	25.5
19	5	0	No	0	5	250	0	0.5	5.5
20	5	0	No	0	5	250	0	0.5	5.5
21	5	0	No	0	5	250	0	0.5	5.5
22	5	0	No	0	5	250	0	0.5	5.5
23	5	0	No	0	5	250	0	0.5	5.5
24	5	0	No	0	5	250	0	0.5	5.5
25	5	3	No	0	2	100	0	0.2	2.2
26	2	0	Yes	5	7	350	20	0.7	27.7
27	7	1	No	0	6	300	0	0.6	6.6
28	6	1	No	0	5	250	0	0.5	5.5
29	5	2	No	0	3	150	0	0.3	3.3
30	3	0	Yes	5	8	400	20	0.8	28.8
									308.5

Conclusion

In this case study, we successfully simulated an inventory management system to optimize the company's ordering process.
By applying a reorder point policy and a fixed order quantity, we were able to monitor the inventory levels, calculate daily costs, and minimize the total expenses associated with holding and ordering products.
The simulation demonstrated how simple decision rules can effectively control inventory operations and reduce unnecessary costs, ensuring better resource utilization and customer satisfaction.

Final Remarks

This simulation highlights the importance of strategic inventory management in any business environment.
By understanding demand patterns and applying systematic policies, companies can avoid overstocking, minimize stockouts, and optimize operational costs.
Future improvements could include integrating more complex demand models, considering lead times, or applying machine learning techniques to predict demand more accurately and further enhance inventory performance.

Prepared by:

Hoda Ahmed Abdel Fattah	*12300462*
Shams Imam Abdel Fattah	*12300326*
Salma Ahmed Mahmoud Abdelgelil	*12300588*

Date: April 27, 2025