ASSIGNMENT NO. 6

Problem Statement: Coffee Shop Line (Simple Queue):
Arrival: Customers arrive at the coffee shop and stand in line. Order Processing: The first customer in line gets their order taken, and the barista starts making the coffee.Serving: Once the first customer is served, they leave the queue, and the next customer in line moves forward to be served. Write a program to implement a simple queue.

CODE:

```
#include <iostream>
#include <string>
using namespace std;

// Node class for each customer
class Node {
public:
    string name;
    Node* next;

    Node(string customerName) {
        name = customerName;
        next = nullptr;
    }
};

class CoffeeShopQueue {
private:
    Node* front;  // Points to the first customer
    Node* rear;   // Points to the last customer

public:
    CoffeeShopQueue() {
        front = rear = nullptr;
    }

    // Check if queue is empty
    bool isEmpty() {
        return (front == nullptr);
    }

    // Enqueue - add customer
    void newCustomer(string name) {
        Node* temp = new Node(name);
```

```
    if (rear == nullptr) {
        // First customer
        front = rear = temp;
    } else {
        rear->next = temp;
        rear = temp;
    }
    cout << name << " joined the line." << endl;
}

// Dequeue - serve customer
void serveCustomer() {
    if (isEmpty()) {
        cout << "No customers in line." << endl;
        return;
    }

    Node* temp = front;
    cout << front->name << "'s order is ready. They leave the line." << endl;

    front = front->next;

    // If queue becomes empty
    if (front == nullptr) {
        rear = nullptr;
    }

    delete temp;
}

// Show current queue
void showQueue() {
    if (isEmpty()) {
        cout << "The line is empty." << endl;
        return;
    }

    cout << "Current Line: ";
    Node* temp = front;
    while (temp != nullptr) {
        cout << temp->name;
        if (temp->next != nullptr) cout << " -> ";
        temp = temp->next;
```

```
        }
        cout << endl;
    }

    // Destructor to free memory
    ~CoffeeShopQueue() {
        while (!isEmpty()) {
            serveCustomer();
        }
    }
};

int main() {
    CoffeeShopQueue shop;
    int option;
    string name;

    do {
        cout << "\n--- Coffee Shop Queue Menu ---" << endl;
        cout << "1. New Customer Arrival (Enqueue)" << endl;
        cout << "2. Serve Customer (Dequeue)" << endl;
        cout << "3. Show Queue" << endl;
        cout << "4. Exit" << endl;
        cout << "Choose an option: ";
        cin >> option;

        switch (option) {
        case 1:
            cout << "Enter customer name: ";
            cin >> name;
            shop.newCustomer(name);
            break;
        case 2:
            shop.serveCustomer();
            break;
        case 3:
            shop.showQueue();
            break;
        case 4:
            cout << "Exiting program..." << endl;
            break;
        default:
            cout << "Invalid option. Try again." << endl;
        }
```

```
    } while (option != 4);

    return 0;
}
```

OUTPUT:

```
--- Coffee Shop Queue Menu ---
1. New Customer Arrival (Enqueue)
2. Serve Customer (Dequeue)
3. Show Queue
4. Exit
Choose an option: 1
Enter customer name: Santosh
Santosh joined the line.

--- Coffee Shop Queue Menu ---
1. New Customer Arrival (Enqueue)
2. Serve Customer (Dequeue)
3. Show Queue
4. Exit
Choose an option: 1
Enter customer name: Ajay
Ajay joined the line.

--- Coffee Shop Queue Menu ---
1. New Customer Arrival (Enqueue)
2. Serve Customer (Dequeue)
3. Show Queue
4. Exit
Choose an option: 1
Enter customer name: Sushant
Sushant joined the line.

--- Coffee Shop Queue Menu ---
1. New Customer Arrival (Enqueue)
2. Serve Customer (Dequeue)
3. Show Queue
4. Exit
Choose an option: 3
Current Line: Santosh -> Ajay -> Sushant
```

--- Coffee Shop Queue Menu ---
1. New Customer Arrival (Enqueue)
2. Serve Customer (Dequeue)
3. Show Queue
4. Exit
Choose an option: 2
Santosh's order is ready. They leave the line.

--- Coffee Shop Queue Menu ---
1. New Customer Arrival (Enqueue)
2. Serve Customer (Dequeue)
3. Show Queue
4. Exit
Choose an option: 3
Current Line: Ajay -> Sushant

--- Coffee Shop Queue Menu ---
1. New Customer Arrival (Enqueue)
2. Serve Customer (Dequeue)
3. Show Queue
4. Exit
Choose an option: 1
Enter customer name: Avinash
Avinash joined the line.

--- Coffee Shop Queue Menu ---
1. New Customer Arrival (Enqueue)
2. Serve Customer (Dequeue)
3. Show Queue
4. Exit
Choose an option: 3
Current Line: Ajay -> Sushant -> Avinash

--- Coffee Shop Queue Menu ---
1. New Customer Arrival (Enqueue)
2. Serve Customer (Dequeue)
3. Show Queue
4. Exit
Choose an option: 2
Ajay's order is ready. They leave the line.

--- Coffee Shop Queue Menu ---
1. New Customer Arrival (Enqueue)

2. Serve Customer (Dequeue)
3. Show Queue
4. Exit
Choose an option: 2
Sushant's order is ready. They leave the line.

--- Coffee Shop Queue Menu ---
1. New Customer Arrival (Enqueue)
2. Serve Customer (Dequeue)
3. Show Queue
4. Exit
Choose an option: 2
Avinash's order is ready. They leave the line.

--- Coffee Shop Queue Menu ---
1. New Customer Arrival (Enqueue)
2. Serve Customer (Dequeue)
3. Show Queue
4. Exit
Choose an option: 2
No customers in line.

--- Coffee Shop Queue Menu ---
1. New Customer Arrival (Enqueue)
2. Serve Customer (Dequeue)
3. Show Queue
4. Exit
Choose an option: 4
Exiting program...