Solution

#include<iostream> using namespace std; struct Order {

String dishName; int orderId; Order\* next;

};

Class Restaurant {

Private:

Order\* front;

Order\* rear;

Public:

Restaurant() {

Front=NULL; rear=NULL;

}

Void enqueueOrder(string dishName, int orderId) { Order\* order=new Order();

Order->dishName=dishName; order->orderId=orderId; order->next=NULL; if(rear==NULL) {

Front=order; rear=order;

} else {

Rear->next=order;

Rear=order;

}

}

Void dequeueOrder() {

If(front!=NULL) {

Order\* temp=front;

Cout << “Order “ << temp->orderId<< “ for “ <<temp-

>dishName<<” processed.”<<endl;

Front=front->next; delete temp;

If(front == NULL) {

Rear=NULL;

}

} else {

Cout<<”NO more order to process”<<endl;

}

}

Void display() {

If (front == NULL) {

Cout << “No orders in the queue.”<<endl; return;

<< “)” << endl;

}

Cout << “Orders in the queue:”<<endl; Order\* temp = front;

While (temp != NULL) {

Cout << temp->dishName << “ (Order ID: “ << temp->orderId

Temp = temp->next;

}

}

};

Int main() {

Restaurant restaurant; restaurant.enqueueOrder(“Pizza”, 1);

Restaurant.enqueueOrder(“Burger”, 2);

Restaurant.enqueueOrder(“Pasta”, 3);

Restaurant.display();

Restaurant.dequeueOrder(); restaurant.dequeueOrder(); restaurant.dequeueOrder(); restaurant.dequeueOrder();

Return 0;

}