# FINAL PROJECT
# COOKING ORDER ORGANIZER

**Members:**

VIÑAS, JUDAH PAULO
BABILA, JOHN RICK
DELIVA, DINCE

## Introduction

The "Cooking Order Organizer" project is a C++ application that manages and organizes cooking orders in a restaurant or culinary setting by utilizing the queueing data structure. Users may enter and queue orders, assuring a first-in, first-out (FIFO) order for efficient kitchen management. It has features like adding new orders, deleting completed orders, and displaying the current order queue. This project may be used in a variety of food service companies to optimize kitchen staff workflow by simplifying the order preparation process and enhancing overall efficiency. The "Cooking Order Organizer" helps to make the culinary operation run more smoothly and efficiently, while also increasing customer satisfaction via fast and precise order fulfillment.

## Algorithm Used

## Queue

**Definition:** A queue is a fundamental data structure in computer science that follows the First-In-First-Out (FIFO) principle. In a queue, elements are added at the rear (enqueue operation) and removed from the front (dequeue operation). This ensures that the element that has been in the queue the longest is the first to be processed or removed. It's similar to the cooking order line in a kitchen, where the first recipe in the queue is the first recipe cooked.

Queue follows **the First In First Out (FIFO) rule - the item that goes in first is the item that comes out first.**

## Enqueue

```
92          case 1:
93              // this is for adding dish to cook
94              cout << "Enter dish name: ";
95              cin >> newOrder.dishName;
96              cout << "For how many person?: ";
97              cin >> newOrder.quantity;
98              enqueue(newOrder);
99              break;
100         case 2:
```

## Dequeue

```
 99                case 2:
100                    // this is for removing dish
101                    if (!isEmpty()) {
102                        dequeue();
103                    } else {
104                        cout << "\n==================================" << endl;
105                        cout << "      ORDER QUEUE IS EMPTY.\n";
106                        cout << "==================================" << endl << endl;
107                    }
108                    break;
109                case 3:
```

## Screenshot of the Program:

```
+---------------------------------+
|     COOKING ORDER ORGANIZER     |
+---------------------------------+
| [1] Add Cooking Order           |
| [2] Serve Cooking Orden         |
| [3] Display Cooking Orders      |
| [4] Exit                        |
+---------------------------------+
Enter your choice (1-4): 12

==================================
    INVALID INPUT. TRY AGAIN!
==================================

+---------------------------------+
|     COOKING ORDER ORGANIZER     |
+---------------------------------+
| [1] Add Cooking Order           |
| [2] Serve Cooking Orden         |
| [3] Display Cooking Orders      |
| [4] Exit                        |
+---------------------------------+
Enter your choice (1-4): 1
Enter dish name: Adobo
For how many person?: 5

=========== ORDER ADDED ===========
 Name : Adobo
 Good for 5 person.
==================================
```

```
+---------------------------------+
|     COOKING ORDER ORGANIZER     |
+---------------------------------+
| [1] Add Cooking Order           |
| [2] Serve Cooking Orden         |
| [3] Display Cooking Orders      |
| [4] Exit                        |
+---------------------------------+
Enter your choice (1-4): 1
Enter dish name: Sinigang
For how many person?: 7

=========== ORDER ADDED ===========
 Name : Sinigang
 Good for 7 person.
==================================

+---------------------------------+
|     COOKING ORDER ORGANIZER     |
+---------------------------------+
| [1] Add Cooking Order           |
| [2] Serve Cooking Orden         |
| [3] Display Cooking Orders      |
| [4] Exit                        |
+---------------------------------+
Enter your choice (1-4): 1
Enter dish name: Inihaw
For how many person?: 2
```

```
========== ORDER ADDED ==========
 Name : Inihaw
 Good for 2 person.
==================================

+--------------------------------+
|     COOKING ORDER ORGANIZER    |
+--------------------------------+
| [1] Add Cooking Order          |
| [2] Serve Cooking Orden        |
| [3] Display Cooking Orders     |
| [4] Exit                       |
+--------------------------------+
Enter your choice (1-4): 1
Enter dish name: Ginataan
For how many person?: 5

========== ORDER ADDED ==========
 Name : Ginataan
 Good for 5 person.
==================================

+--------------------------------+
|     COOKING ORDER ORGANIZER    |
+--------------------------------+
| [1] Add Cooking Order          |
| [2] Serve Cooking Orden        |
| [3] Display Cooking Orders     |
| [4] Exit                       |
+--------------------------------+
Enter your choice (1-4): 1
```

```
+--------------------------------+
|     COOKING ORDER ORGANIZER    |
+--------------------------------+
| [1] Add Cooking Order          |
| [2] Serve Cooking Orden        |
| [3] Display Cooking Orders     |
| [4] Exit                       |
+--------------------------------+
Enter your choice (1-4): 3

========== ORDER LIST ===========
 [1] Adobo
     Good for 5 person.
 [2] Sinigang
     Good for 7 person.
 [3] Inihaw
     Good for 2 person.
 [4] Ginataan
     Good for 5 person.
==================================

+--------------------------------+
|     COOKING ORDER ORGANIZER    |
+--------------------------------+
| [1] Add Cooking Order          |
| [2] Serve Cooking Orden        |
| [3] Display Cooking Orders     |
| [4] Exit                       |
+--------------------------------+
Enter your choice (1-4): 2
```

```
+--------------------------------+
|     COOKING ORDER ORGANIZER    |
+--------------------------------+
| [1] Add Cooking Order          |
| [2] Serve Cooking Orden        |
| [3] Display Cooking Orders     |
| [4] Exit                       |
+--------------------------------+
Enter your choice (1-4): 1
Enter dish name: Gulay
For how many person?: 5

==================================
      ORDER QUEUE IS FULL.
==================================

+--------------------------------+
|     COOKING ORDER ORGANIZER    |
+--------------------------------+
| [1] Add Cooking Order          |
| [2] Serve Cooking Orden        |
| [3] Display Cooking Orders     |
| [4] Exit                       |
+--------------------------------+
Enter your choice (1-4): 3

========== ORDER LIST ===========
 [1] Adobo
     Good for 5 person.
```

```
| [3] Display Cooking Orders     |
| [4] Exit                       |
+--------------------------------+
Enter your choice (1-4): 2

========= ORDER COMPLETED =========
 Name: Adobo
 Good for 5 person.
==================================

+--------------------------------+
|     COOKING ORDER ORGANIZER    |
+--------------------------------+
| [1] Add Cooking Order          |
| [2] Serve Cooking Orden        |
| [3] Display Cooking Orders     |
| [4] Exit                       |
+--------------------------------+
Enter your choice (1-4): 2

========= ORDER COMPLETED =========
 Name: Sinigang
 Good for 7 person.
==================================

+--------------------------------+
|     COOKING ORDER ORGANIZER    |
+--------------------------------+
```

```
+---------------------------------+
|     COOKING ORDER ORGANIZER     |
+---------------------------------+
| [1] Add Cooking Order           |
| [2] Serve Cooking Orden         |
| [3] Display Cooking Orders      |
| [4] Exit                        |
+---------------------------------+
Enter your choice (1-4): 3

=========== ORDER LIST ============
 [1] Inihaw
     Good for 2 person.
 [2] Ginataan
     Good for 5 person.
===================================

+---------------------------------+
|     COOKING ORDER ORGANIZER     |
+---------------------------------+
| [1] Add Cooking Order           |
| [2] Serve Cooking Orden         |
| [3] Display Cooking Orders      |
| [4] Exit                        |
+---------------------------------+
Enter your choice (1-4): 2

========= ORDER COMPLETED =========
 Name: Inihaw
 Good for 2 person.
===================================
```

```
+---------------------------------+
|     COOKING ORDER ORGANIZER     |
+---------------------------------+
| [1] Add Cooking Order           |
| [2] Serve Cooking Orden         |
| [3] Display Cooking Orders      |
| [4] Exit                        |
+---------------------------------+
Enter your choice (1-4): 2

========= ORDER COMPLETED =========
 Name: Ginataan
 Good for 5 person.
===================================

+---------------------------------+
|     COOKING ORDER ORGANIZER     |
+---------------------------------+
| [1] Add Cooking Order           |
| [2] Serve Cooking Orden         |
| [3] Display Cooking Orders      |
| [4] Exit                        |
+---------------------------------+
Enter your choice (1-4): 3

===================================
      ORDER QUEUE IS EMPTY.
===================================
```

```
===================================
      ORDER QUEUE IS EMPTY.
===================================

+---------------------------------+
|     COOKING ORDER ORGANIZER     |
+---------------------------------+
| [1] Add Cooking Order           |
| [2] Serve Cooking Orden         |
| [3] Display Cooking Orders      |
| [4] Exit                        |
+---------------------------------+
Enter your choice (1-4): 2

===================================
      ORDER QUEUE IS EMPTY.
===================================

+---------------------------------+
|     COOKING ORDER ORGANIZER     |
+---------------------------------+
| [1] Add Cooking Order           |
| [2] Serve Cooking Orden         |
| [3] Display Cooking Orders      |
| [4] Exit                        |
+---------------------------------+
Enter your choice (1-4): 4

===================================
       EXITING THE PROGRAM
===================================
```

## Source Code

```
#include <iostream>
#define MAX 4
using namespace std;

struct Order {
   string dishName;
   int quantity;
};

Order orderQueue[MAX];
int rear = -1;
int front = 0;
int count = 0;

int isFull() {
   return count == MAX;
}

void enqueue(Order newOrder) {
   if (!isFull()) {
      if (rear == MAX - 1) {
         rear = -1;
      }
      orderQueue[++rear] = newOrder;
      count++;
      cout << "\n========== ORDER ADDED ==========" << endl;
      cout << " Name : " << newOrder.dishName << endl;
      cout << " Good for " << newOrder.quantity << " person. " << endl;
      cout << "==================================" << endl << endl;
   } else {
      cout << "\n==================================" << endl;
      cout << "      ORDER QUEUE IS FULL." << endl;
      cout << "==================================" << endl << endl;
   }
}


int isEmpty() {
   return count == 0;
}

Order dequeue() {
   Order completedOrder = orderQueue[front++];
   if (front == MAX) {
      front = 0;
   }
   count--;
   cout << "\n========= ORDER COMPLETED =========" << endl;
   cout << " Name: " << completedOrder.dishName <<endl;
   cout << " Good for " << completedOrder.quantity << " person." << endl;
   cout << "==================================" << endl << endl;
   return completedOrder;
}

// this is for displaying order queue
void displayOrders() {
   if (isEmpty()) {
```

```
        cout << "\n==================================" << endl;
        cout << "      ORDER QUEUE IS EMPTY." << endl;
        cout << "==================================" << endl << endl;
        return;
    }

    int i = front;
    cout << "\n========== ORDER LIST ==========" << endl;
    for (int j = 0; j < count; ++j) {
        cout << " [" << j+1 << "] " << orderQueue[i].dishName << endl;
        cout << "     Good for " << orderQueue[i].quantity << " person." << endl;
        i = (i + 1) % MAX;
    }
    cout << "==================================" << endl << endl;
}

int main() {
    int choice;
    Order newOrder;

    do {
        cout << "+--------------------------------+\n";
        cout << "|     COOKING ORDER ORGANIZER     |\n";
        cout << "+--------------------------------+\n";
        cout << "| [1] Add Cooking Order          |\n";
        cout << "| [2] Serve Cooking Orden        |\n";
        cout << "| [3] Display Cooking Orders     |\n";
        cout << "| [4] Exit                       |\n";
        cout << "+--------------------------------+\n";
        cout << "Enter your choice (1-4): ";
        cin >> choice;

        switch (choice) {
            case 1:
                // this is for adding dish to cook
                cout << "Enter dish name: ";
                cin >> newOrder.dishName;
                cout << "For how many person?: ";
                cin >> newOrder.quantity;
                enqueue(newOrder);
                break;
            case 2:
                // this is for removing dish
                if (!isEmpty()) {
                    dequeue();
                } else {
                    cout << "\n==================================" << endl;
                    cout << "      ORDER QUEUE IS EMPTY.\n";
                    cout << "==================================" << endl << endl;
                }
                break;
            case 3:
                displayOrders();
                break;
            case 4:
                cout << "\n==================================" << endl;
                cout << "       EXITING THE PROGRAM \n";
                  cout << "==================================" << endl << endl;
```

```
            break;
        default:
            cout << "\n===================================" << endl;
            cout << "    INVALID INPUT. TRY AGAIN!\n";
            cout << "===================================" << endl << endl;
    }
  } while (choice != 4);

  return 0;
}
```