ARBAMINCH UNIVERSITY INSTITUTE OF TECHNOLOGY

FACULTY OF COMPUTING AND SOFTWARE ENGINEERIN

COURSE NAME : FUNDAMENTALS OF DATA STRUCTURE AND ALGORITHM

COURSE CODE :sEng2122

SECTION B

Name	**ID**
Yohana Gezahegne	NSR/1051/16
Bezawit Melese	NSR/182/16
Rawda Oumer	NSR/813/16
Rediet Girma	NSR/809/16

Submitted to : Yitayish Lema

Submitted Date : June 11,2025

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

#define MAX_SIZE 100

class Queue {
private:
    int arr[MAX_SIZE];
    int front;
    int rear;
    int size;

public:
    Queue() {
        front = -1;
        rear = -1;
        size = 0;
    }

    bool isEmpty() {
        return size == 0;
    }

    bool isFull() {
        return size == MAX_SIZE;
```

```
}

void enqueue(int value) {
    if (isFull()) {
        cout << "Queue is full! Cannot enqueue " << value << endl;
        return;
    }

    if (isEmpty()) {
        front = 0;
    }

    rear = (rear + 1) % MAX_SIZE;
    arr[rear] = value;
    size++;
    cout << value << " enqueued to queue" << endl;
}

int dequeue() {
    if (isEmpty()) {
        cout << "Queue is empty! Cannot dequeue" << endl;
        return -1;
    }

    int value = arr[front];
    if (front == rear) {
        front = -1;
```

```
            rear = -1;
        } else {
            front = (front + 1) % MAX_SIZE;
        }
        size--;
        cout << value << " dequeued from queue" << endl;
        return value;
    }

    void display() {
        if (isEmpty()) {
            cout << "Queue is empty!" << endl;
            return;
        }

        cout << "Queue elements: ";
        int count = 0;
        int index = front;
        while (count < size) {
            cout << arr[index] << " ";
            index = (index + 1) % MAX_SIZE;
            count++;
        }
        cout << endl;
    }
};
```

```
int main() {
    Queue q;
    int choice, value;

    do {
        cout << "\nQueue Operations Menu:\n";
        cout << "1. Enqueue\n";
        cout << "2. Dequeue\n";
        cout << "3. Display\n";
        cout << "4. Exit\n";
        cout << "Enter your choice: ";
        cin >> choice;

        switch (choice) {
            case 1:
                cout << "Enter value to enqueue: ";
                cin >> value;
                q.enqueue(value);
                break;
            case 2:
                q.dequeue();
                break;
            case 3:
                q.display();
                break;
            case 4:
                cout << "Exiting program" << endl;
```

```
            break;
        default:
            cout << "Invalid choice! Please try again." << endl;
      }
   } while (choice != 4);

   return 0;
}
```