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

#define MAX 1000

class Queue {
private:
   int front, rear, size;
   int queue[MAX];

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


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


   bool isFull() {
      return size == MAX;
   }
```

```
void enqueue(int x) {
    if (isFull()) {
        cout << "Queue is full, cannot enqueue " << x << endl;
        return;
    }
    if (rear == -1) {
        front = rear = 0;
    } else {
        rear = (rear + 1) % MAX;
    }
    queue[rear] = x;
    size++;
    cout << x << " enqueued to queue" << endl;
}


int dequeue() {
    if (isEmpty()) {
        cout << "Queue is empty, cannot dequeue" << endl;
        return -1;
    }
    int item = queue[front];
    if (front == rear) {
        front = rear = -1;
    } else {
        front = (front + 1) % MAX;
    }
```

```
        size--;
        return item;
    }


    int peek() {
        if (isEmpty()) {
            cout << "Queue is empty" << endl;
            return -1;
        }
        return queue[front];
    }


    void display() {
        if (isEmpty()) {
            cout << "Queue is empty" << endl;
            return;
        }
        cout << "Queue elements: ";
        for (int i = 0; i < size; i++) {
            cout << queue[(front + i) % MAX] << " ";
        }
        cout << endl;
    }
};

int main() {
    Queue q;
    q.enqueue(1);
    q.enqueue(2);
    q.enqueue(3);
    q.enqueue(4);

    q.display();

    cout << q.dequeue() << " dequeued from queue" << endl;

    q.display();

    cout << "Front element is: " << q.peek() << endl;

    return 0;
}
```