

RajvaibhavRahane  
17u283 223045  
SE-C Comp,Viit,Pune

---

**CODE:**

```
/*
 *
 *@RajvaibhavRahane
 */
#include<iostream>
#include<exception>
using namespace std;
class FullQueueException:public exception{
    public:
        const char* what()const throw(){
            return "FullQueueException";
        }
};
class EmptyQueueException:public exception{
    public:
        const char*what()const throw(){
            return "EmptyQueueException";
        }
};
#define SIZE 5
class Queue{
    private:
        int *arr;
        int elements;
        inthead,tail;
    public:
        Queue(){
            arr=new int[SIZE];
            elements=0;
            head=tail=0;
        }~Queue(){
            delete []arr;arr=NULL;
            cout<<"Queue Destructed\n";
        }
        bool isEmpty(){
            return elements==0;
        }
        bool isFull(){
            return elements==SIZE;
        }
        intdeQueue();
        void enQueue(int element);
        void printQueue();
};
void Queue::enQueue(int element){
    if(!isFull()){
        arr[tail]=element;
        tail=(tail+1)%SIZE;
        elements++;
    }else throw FullQueueException();
}
```

```

int Queue::deQueue(){
    if(!isEmpty()){
        int element=arr[head];
        head=(head+1)%SIZE;
        elements--;
        return element;
    }else throw EmptyQueueException();
}

void Queue::printQueue(){
    int printer=head,iterations=elements;
    if(iterations!=0)
        while(iterations--){
            cout<<arr[printer]<<" ";
            printer=(printer+1)%SIZE;
        }
    else cout<<"Empty Queue";
    cout<<endl;
}

void printMenu(){
    cout<<"1)EnQueue element\t";
    cout<<"2)DeQueue element\t";
    cout<<"3)Print Queue\t";
    cout<<"4)Exit\nChoice : ";
}

int main(){
    {
        Queue q;
        int choice,element;
        printMenu();
        do{
            cout<<endl;cin>>choice;
            switch(choice){
                case 1:{
                    cout<<"Enter Task No ";cin>>element;
                    try{
                        q.enqueue(element);
                    }catch(FullQueueException& e){
                        cout<<e.what()<<" was thrown\n";
                    }break;
                }
                case 2:{
                    try{
                        element = q.deQueue();
                        cout<<"Task "<<element<<" was deQueued\n";
                    }catch(EmptyQueueException& e){
                        cout<<e.what()<<" was thrown\n";
                    }
                    break;
                }
                case 3:{
                    q.printQueue();break;
                }
                case 4:{break;}
            }
        }while(choice!=4);
    }
}

```

```
    return 0;  
}
```

---

### Output:

```
rajrahane@visraj-lenovo-g500: ~/Desktop/c++/Labs/FDS/Queue  
rajrahane@visraj-lenovo-g500:~/Desktop/c++/Labs/FDS/Queue$ g++ -o queueUsingArray queueUsingArray.cpp  
rajrahane@visraj-lenovo-g500:~/Desktop/c++/Labs/FDS/Queue$ ./queueUsingArray  
1)EnQueue element      2)DeQueue element      3)Print Queue      4)Exit  
Choice :  
1 47 1 48 1 49 1 45 1 46 1 42  
Enter Task No  
Enter Task No  
Enter Task No  
Enter Task No  
Enter Task No  
Enter Task No FullQueueException was thrown  
3:  
47 48 49 45 46  
2 1 42  
Task 47 was deQueued  
Enter Task No  
2 1 43  
Task 48 was deQueued  
Enter Task No  
3  
49 45 46 42 13  
2 2 2 2 2 2  
Task 49 was deQueued  
Task 45 was deQueued  
Task 46 was deQueued  
Task 42 was deQueued  
Task 13 was deQueued  
EmptyQueueException was thrown  
4  
Queue Destructed  
rajrahane@visraj-lenovo-g500:~/Desktop/c++/Labs/FDS/Queue$
```