



CL-2001 Data Structures Lab # 8

Objectives:

- Queue ADT
- Enqueue, Dequeue
- Rear, Front
- Queue Linked List

Note: Carefully read the following instructions (*Each instruction contains a weightage*)

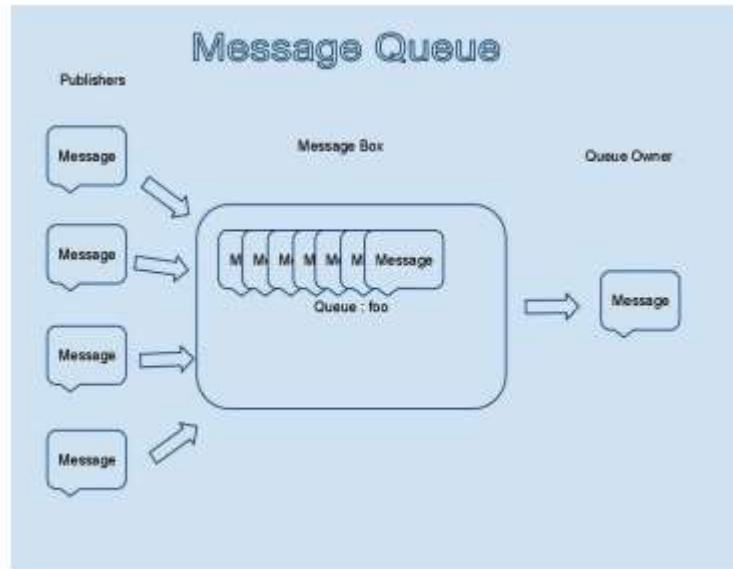
1. There must be a block of comments at start of every question's code by students; the block should contain brief description about functionality of code.
2. Comment on every function and about its functionality.
3. Mention comments where necessary such as comments with variables, loop, classes etc to increase code understandability.
4. Use understandable name of variables.
5. Proper indentation of code is essential.
6. Write a code in C++ language.
7. Make a Microsoft Word file and paste all of your C++ code with all possible screenshots of every task **outputs in Microsoft Word and submit word file. Do not submit .cpp file.**
8. First think about statement problems and then write/draw your logic on copy.
9. After copy pencil work, code the problem statement on MS Studio C++ compiler.
10. At the end when you done your tasks, attached C++ created files in MS word file and make your submission on Google Classroom. (Make sure your submission is completed).
11. Please submit your file in this format **20F1234_L4**.
12. **Do not submit your assignment after deadline. Late and email submission is not accepted.**
13. **Do not copy code from any source otherwise you will be penalized with negative marks.**

Problem: 1 |

Implement the priority queue using Array Based for 10 elements.

Problem: 2 |

Messenger is used to send or receive text messages. When someone is offline a messenger maintains a buffer of messages which is delivered to the receiver when he gets online. The phenomena take place on simple timestamp phenomena, the message delivered earlier will be sent to the receiver first and the message received late will be delivered after it. Sometime a message in the buffer may have higher priority so it should be delivered earlier on the higher priority. Some of the messages are to be delivered on a particular day or a date are also in the same buffer. Your task is to select a suitable data structure and implement the requirements mentioned above.



You need to implement program using linked list which shows a user to be offline, buffer the messages, with a click or a key stroke make user online and deliver the messages according to the mentioned criteria.

Problem: 3 |

Implement double ended queue Array based implementation. Please implement the functions below

```

#include<iostream>
using namespace std;
class DEQueue
{
    int* arr;
    int front, rear;
public:
    DEQueue(int size)
    {
        arr = new int[size];
        front = -1;
        rear = -1;
    }
    void push_front(int);
    void push_back(int);
    void pop_front();
    void pop_back();
    int get_front();
  
```



```
int get_back();
bool full();
bool empty();
};
int main() {
    DEQueue obj(10);
    system("pause");
    return 0;
}
```

Problem: 4 |

Implement double ended queue using Linked List



Best of luck