University of Central Punjab

Faculty of Information Technology

Data Structures and Algorithms
Spring 2021

Lab 05	
Topic	• Abstract Classes • Templates • Queues
Objective	The basic purpose of this lab is to implement ADT of queue and test its applications.

Instructions:

- Indent your code.
- Comment your code.
- Use meaningful variable names.
- Plan your code carefully on a piece of paper before you implement it.
- Name of the program should be same as the task name. i.e. the first program should be Task_1.cpp
- **void main() is not allowed. Use int main()**
- **You have to work in multiple files. i.e separate .h and .cpp files**
- **You are not allowed to use system("pause")**
- **You are not allowed to use any built-in functions**
- **You are required to follow the naming conventions as follow:**
    - **Variables:** firstName; (no underscores allowed)
    - **Function:** getName(); (no underscores allowed)
    - **ClassName:** BankAccount (no underscores allowed)

Students are required to complete the following tasks in lab timings.

Task 1

Queue: Queue is a data structure that works on First in First out (FIFO) approach. In general, items are added to the end of Array (In the way that we join at the end of a queue for a bus) and items are removed from the front of an Array. (The people at the front of the queue for the bus can get on the bus first.)

Attributes:

Type*arr;
int front;
int rear;
int maxSize;

bool empty() : Returns whether the **Queue** is empty or not. Time Complexity should be: O(1)

bool full() : Returns whether the **Queue** is full or not. Time Complexity should be: O(1)
int size() : Returns the current size of the **Queue**.
Type front () : Returns the front element of the **Queue.**

void enqueue(Type) : Adds the element of type Type at the end of the **Queue**.

Type dequeue() : Deletes the first most element of the **Queue** and returns it.

void rearrange(Type) : rearranges the queue as one dequeue happens this function will be called.

void showQueue (Queue <Type> obj): a global function show **Queue** which should display all the contents of the **Queue**. Complexity should be: O(n)

- Write non-parameterized constructor for the above class.
- Write Copy constructor for the above class.
- Write Destructor for the above class.

NOTE: Use an array of fixed size in this task do not do any regrow.

Task 2

Write a program to sort the elements present in the queue make a sortQueue() function. Use a display() function to display the elements of the queue.

Task 3

You are required to add two numbers of 10 digits each using Stack and Queue. You may use the Stack implementation from the last lab. You may follow the following steps to solve this problem unless you can think of a better method!

1. Separate the digits of each number and insert them into two separate queues starting from their last digits.

2. Dequeue one number each from the two queues.

3. Add them.

4. Divide the sum by 10.

5. Push the remainder into a Stack and the Quotient (i.e., carry) would be added to the sum of next two numbers.

6. Repeat Steps 2 – 5 until both queues are empty.

7. Pop all elements of the Stack one by one and push them onto a new Stack.

8. Print the new Stack. It will be the sum of the two 10-digit numbers.