# IIT Ropar
# CSL201 Data Structures
# Semester 1, AY 2018/19

Lab Assignment 2 - 40 marks Due on September 11, 11:59 PM

---

## Objective

The objective is to learn to implement Stacks and Queues.

## Instructions

1. You are to use C++ programming language to complete the assignment.
2. Provide a Makefile to compile your final code.
3. All function and class declarations should be in ".hpp" files while the definitions should be in ".cpp" files. It is recommended that you have one header file and one cpp file for each class. If the class is templated, declare it is the hpp file itself.
4. This is an individual assignment. You can have high level discussions with other students, though.
5. Include a "Readme.txt" file on how to compile and run the code.
6. Upload your submission to moodle by the due date and time.
7. Late submission not allowed. No assignment will be accepted through email after submission system closes. I suggest you to upload the assignment at least 1 hour in advance.
8. If you find assignment description is vague, take appropriate assumption and write that in the Readme. Clearly state assumptions and tell those assumptions during viva.
9. If any student asks for deadline extension or sends assignment through email, he or she will get 5% penalty.
10. I should be able to use the class you have implemented in my program the way we use STL classes. There will be a master program to test the classes.

## Program Description

In this assignment, you need to design and implement a special templated queue abstract data type (ADT). It should support all the standard functions of a queue. In addition, it should also support a special function min() in constant time, i.e. O(1). This function returns reference of the minimum element. You are allowed to use constant amount of extra memory. Note that enqueue and dequeue should also be O(1). Use linked-list based implementation.

The queue should be implemented such that the vegetable and fruit items of assignment 1 can also be added to the queue. You are allowed to use the classes you designed in the previous assignment for this. To compare two fruits or vegetables, you can use their calorie value (to implement min function). You may need to write an overloaded comparison function. You are allowed to take this as a global function.

The interface of the modified queue is given in file sQueue.hpp.

Your main program should take the fruits.txt or vegetables.txt as input, and depending the the type of input file, it should add all the items to the the queue (using enqueue function). The interface should allow users to remove elements from the queue and view minimum.