University of Newcastle
Discipline of Computing and Information Technology
Semester 2, 2018 - SENG1120/6120

Assignment 2

Due using the Blackboard Assignment submission facility:
11:59PM – October 19th, 2018

NOTE: *The important information about submission and code specifics at the end of this assignment specification***.**

INTRODUCTION

In lectures, we have discussed the use of `templates` to provide the compiler with blueprints for functions and classes. These are used by the compiler to produce code that implements functions and/or classes that are suitable for the type(s) to which you apply them in your program code.

ASSIGNMENT TASK

Consider the situation where you are working for a software company that will develop a reporting tool for school administrators. One of the functionalities of that tool is to count the number of fails (FF), passes (P), credits (C), distinctions (D) and high distinctions (HD) among the students. In addition, the program should be able to calculate the average, minimum and maximum scores, and standard deviation. You will be tasked with developing some of the classes for that tool. Standard deviation (σ) can be calculated as:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

, where N is the number of samples, x_i is sample i, and μ is the average of the samples.

For more info, please visit: `https://www.mathsisfun.com/data/standard-deviation.html`

The counting is done using a **queue** that stores the marks, and the marks in each grade band are stored in individual **stacks**, one for each grade. You will be provided a main program that will do the counting, and you will be required to create both `Queue` and `Stack` classes (based on the `LinkedList` and `Node` classes from assignment 1) that interact with it. The average, minimum, maximum and standard deviation calculations must be done in a method inside the class `Queue`.

All classes that you design and submit need to use templates (i.e. `Queue`, `Stack`, `LinkedList` and `Node` will be class templates).

SENG6120 students (or for bonus marks): Note that the grades in the main program provided are ordered, which means the stacks that store the marks in each grade band will also be ordered. SENG6120 students need to change the main program so that even if the variable `vectorMarks` is not ordered, the five resulting stacks will be. SENG1120 students who implement this feature correctly will get a 1-mark bonus. You are NOT ALLOWED to use the `order()` method from LinkedList. The challenge is to implement `sort()` within the `Queue` class, so that the sorting in done by dequeuing and enqueuing elements from/into the queue `marks`.

SUBMISSION

Make sure your code works with the files supplied, and DO NOT change them. For marking, we will add the main file `GradesDemo.cpp` to the project and compile it using the `makefile`, together with your own files. If it does not compile or run, your mark will be zero.

Your submission should be made using the Assignments section of the course Blackboard site. **Incorrectly submitted assignments will not be marked.** You should provide the `.h` and `.template` files related to the `Queue`, `Stack`, `LinkedList` and `Node` classes. Also, if necessary, provide a `readme.txt` file containing any instructions for the marker. Each program file should have a proper header section including your name, course and student number, and your code should be properly documented.

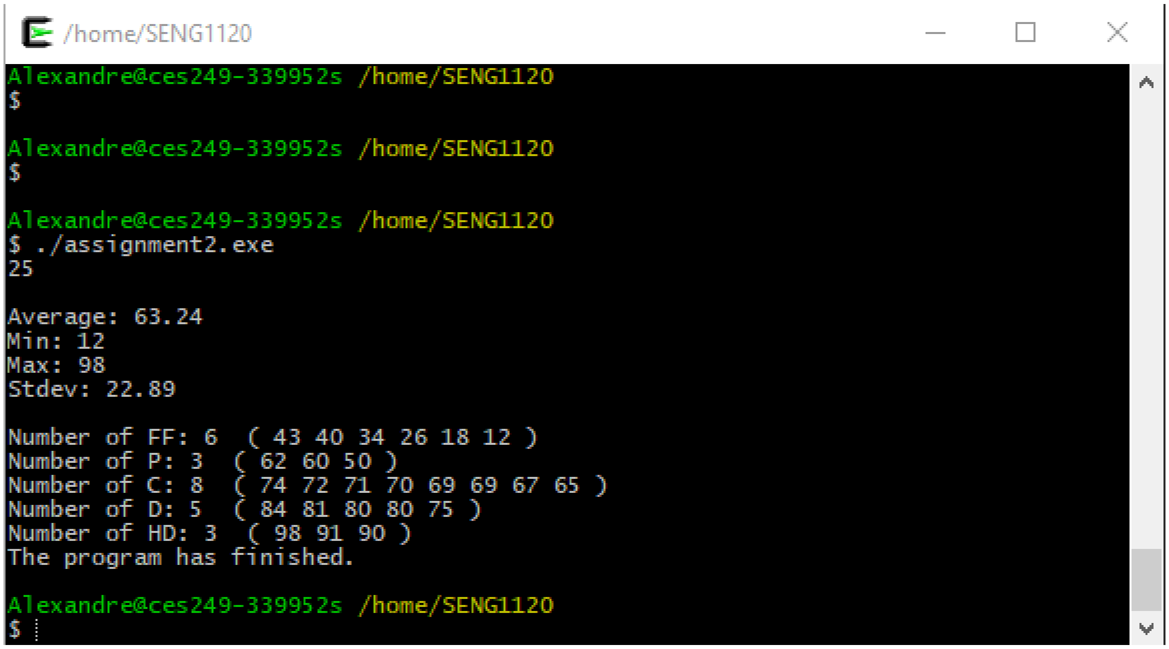
Remember that your code should compile and run correctly using Cygwin. There should be no segmentation faults or memory leaks during or after the execution of the program.

Compress all your files, including the cover sheet, into a single .zip file and submit it in by clicking in a link that will be created in the Assignments section on Blackboard.

Late submissions are subjected to the rules specified in the Course Outline. Finally, a completed Assignment Cover Sheet should accompany your submission.

This assignment is worth 10 marks of your final result for the course.

Compiling and running your files together with the demo file provided should output the following result:



```
Alexandre@ces249-339952s /home/SENG1120
$

Alexandre@ces249-339952s /home/SENG1120
$

Alexandre@ces249-339952s /home/SENG1120
$ ./assignment2.exe
25

Average: 63.24
Min: 12
Max: 98
Stdev: 22.89

Number of FF: 6  ( 43 40 34 26 18 12 )
Number of P: 3  ( 62 60 50 )
Number of C: 8  ( 74 72 71 70 69 69 67 65 )
Number of D: 5  ( 84 81 80 80 75 )
Number of HD: 3  ( 98 91 90 )
The program has finished.

Alexandre@ces249-339952s /home/SENG1120
$
```

```
Alexandre@ces249-339952s /home/SENG1120
$ ./assignment2.exe
25

Average: 63.24
Min: 12
Max: 98
Stdev: 22.89

Number of FF: 6  ( 43 40 34 26 18 12 )
Number of P: 3  ( 62 60 50 )
Number of C: 8  ( 74 72 71 70 69 69 67 65 )
Number of D: 5  ( 84 81 80 80 75 )
Number of HD: 3  ( 98 91 90 )
The program has finished.
```