Lab-4 Notes

Overview: You need to implement an elevator management program, which takes care of people getting into the elevator and out of elevator in an organized manner.

Here is some rules:

Ground floor rules(Queue-like structure):

- The first person in line will be the first to get on the elevator (Queue-like structure)
- There is no limit to the number of persons getting in the waiting line.

Elevator Rules (Stack-like structure):

- The first person to get into the elevator will be the last one to get off
- Only 7 people can be in the elevator at the same time.

For completing this task, you have been given one input.txt file, which consists of a set of commands.

WAIT <name> - Someone is waiting in line for the elevator
- no white space, only single word in name

PICK_UP - This instruction determines the persons to be put in the elevator according to the number of available paces

DROP_OFF <#> - # number of people to get out of elevator
- After that elevator returns to the ground floor.

INSPECTION - Print the current status of the elevator, like below:

```
 Elevator status:

The elevator is not empty.
Mark will be the next person to leave the elevator.
Stacy will be the next person to get on the elevator.
```

There is no EXIT command in the input.txt file. So you have to read until the end of the file, that you can do with:

```
while(infFileObject >> someVar){
   // reading from the file

}
```

- Example of input.txt file has been given on the wiki page.

Technical Part:

For completing the task, you have been given two interfaces.

1. Stack Interface
2. Queue Interface

There is also a very good explanation regarding inheritance on wiki page by Dr. Gibbons

1. Stack Inheritance: Every pure virtual method with a virtual keyword, you have to have implement in the derived class.

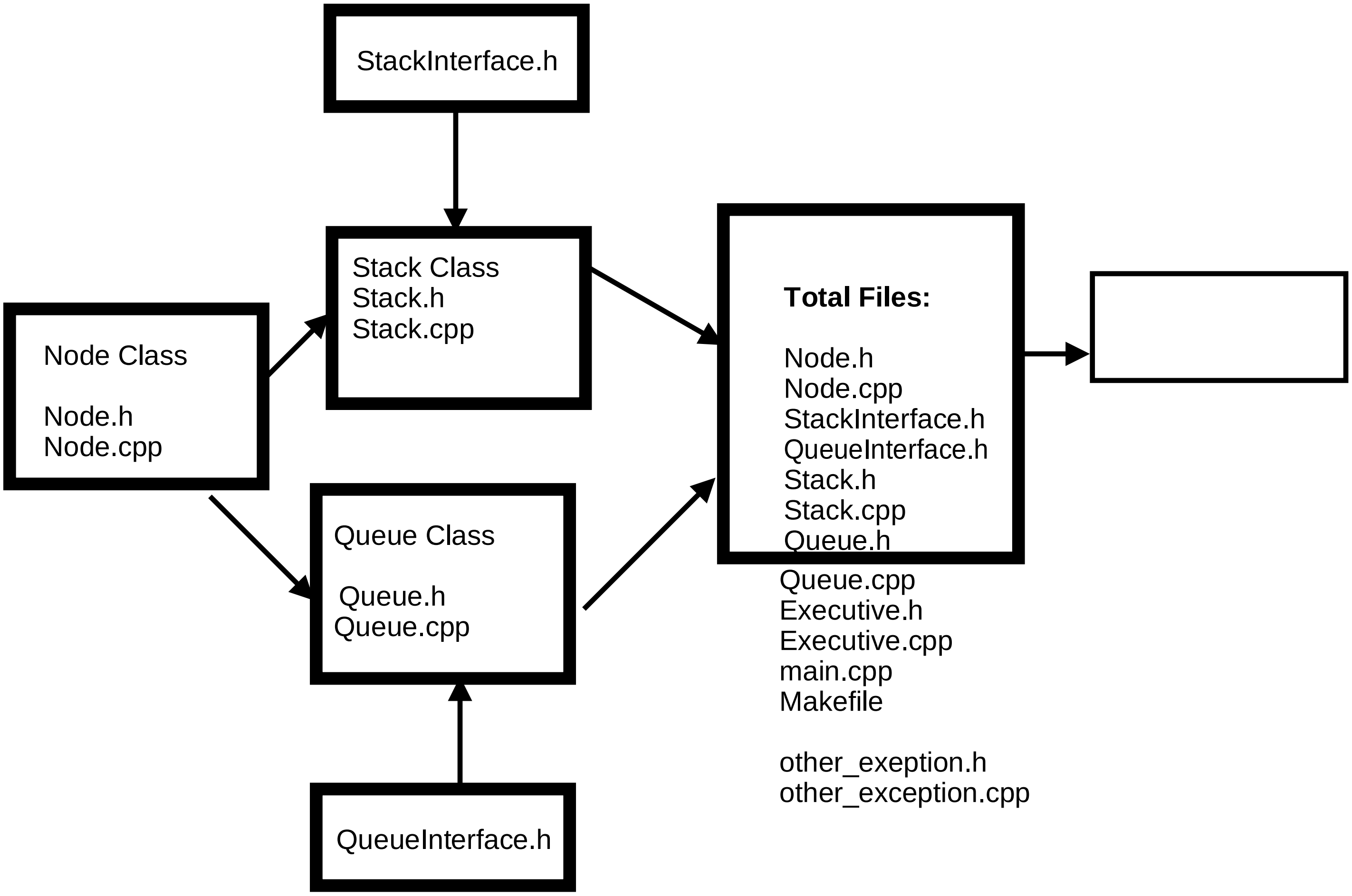
   - The number of methods and explanations are also given on the wiki page.
   - There is no StakcInterface.cpp. You just need to write StackInterface.h. The method implementation would be in derived class Stack.

2. Queue Inheritance: Every pure virtual method with a virtual keyword, you have to have implement in the derived class.

   - The number of methods and explanations are also given on the wiki page.
   - There is no QueueInterface.cpp. You just need to write only QueueInterface.h. The method implementations would be in derived class Queue.

- Now you do not need to make a class which inherits std::runtime_error. Just throw the exception as you did in previous labs. (refer to the Wiki page for details)

Here is the Structure of the lab:


StackInterface.h
Stack Class
Stack.h
Stack.cpp
Node Class
Node.h
Node.cpp
Queue Class
Queue.h
Queue.cpp
QueueInterface.h
Total Files:
Node.h
Node.cpp
StackInterface.h
QueueInterface.h
Stack.h
Stack.cpp
Queue.h
Queue.cpp
Executive.h
Executive.cpp
main.cpp
Makefile
other_exeption.h
other_exception.cpp