The Conga Line lab

CompSys.202 / MechEng.270

Introduction

This lab will create a singly linked-list. However, it is not a traditional singly linked-list. We customise it to simulate a Conga line. To the most part, it behaves like a singly linked-list in that:

- We have an internal type (similar to Node), only here it is called **Participant**
- The **CongaLine** has a reference to the head and tail, only here we call it **frontPerson** and **lastPerson**
- Each Participant has a pointer to the next Participant, just like a Node has a pointer to the next Node
- Participants can **joinIn()**, much like we can add() new Node to a linked-list
- We can **printCongaLine()** and determine **numberOfParticipants()**

The differences with the CongaLine from a standard linked-list, is that we allow additional rules and behaviour:

- We can **makeCircle()**, where the front Participant joins the last one Participant to form a circle
- We can **breakCircle()** so it becomes a line again, where the front person stops holding the last
- We also do not allow someone to **joinIn()** if the Conga line **isCompleteCircle()**

The code provided

You are given 5 files:

CongaLine.h Header file declaring a Conga line class (similar to a LinkedList)

CongaLine.cpp Definition of the Conga line class

Participant.h Header fine declaring a Participant class (similar to Node)

Participant.cpp Definition of the Participant class

main.cpp: The main testing file. Creates a Conga line and prints its state

What to do

You need to only modify CongaLine.cpp, by completing the empty methods with "todo" comments. Make sure you read the comments clearly, as they help you with hints. The statements inside main.cpp will also help you determine if your code is completed correctly. The expected output of your program is as follows:

```
Participants in line: 2
Made circle with only 2 people? 0
(Bob) ==> (Amy)

(Cindy) ==> (Bob) ==> (Amy)                          // represents a line
... ==> (Cindy) ==> (Bob) ==> (Amy) ==> ...          // represents a circle

Could Doug join in a completed circle? 0
... ==> (Cindy) ==> (Bob) ==> (Amy) ==> ...

Participants in circle: 4
... ==> (Doug) ==> (Cindy) ==> (Bob) ==> (Amy) ==> ...
```