

Assignment Sheet

Course, instructor name & contact info	GAME255 – Data Structures and Design Patterns Jean – Paul Amore, jean-paul.amore@humber.ca
Assignment name	Lab 2 – Josephus Position Using a Double Circular Linked List
Grade value	6% (Rubrics attached)
Due date	Week 4
Individual or group assignment	Individual. You may consult with your peers on this assignment, but you are not allowed to share your work or submit work that is not yours. Violation of this may result in an academic misconduct penalty.
Submission instructions	Submit your C++ file Main.cpp on Blackboard
Targeting these learning outcomes from course outline	<ul style="list-style-type: none">Describe the uses of the common data structures in games.

Assignment Instructions

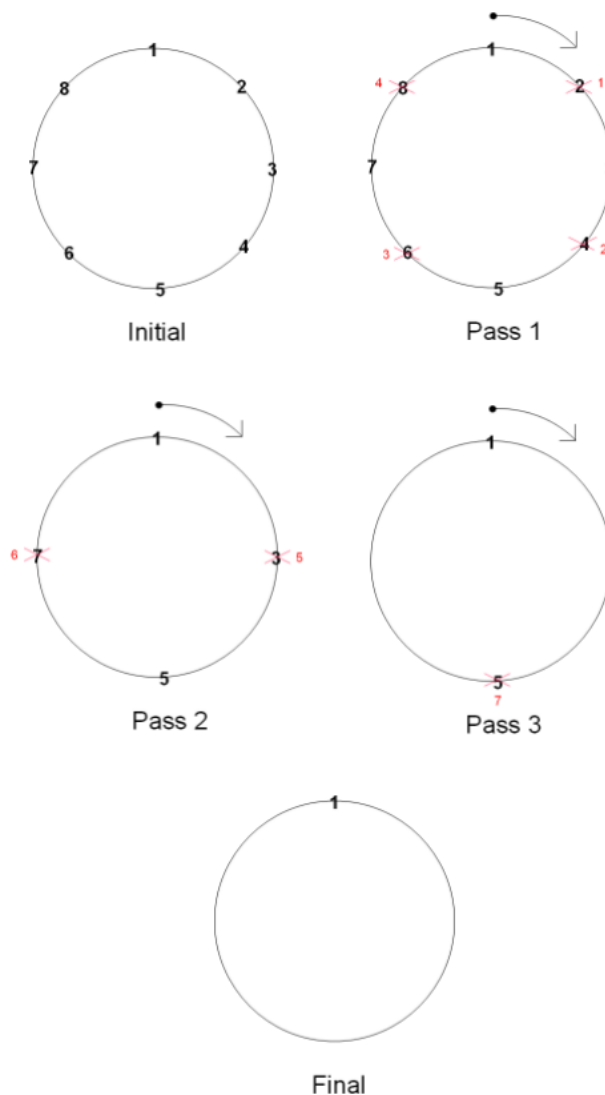
Using the provided project solution, complete the implementation for the member function `void CDList<T>::getJosephusPosition(int)`.

The Josephus Position assumes there are n people standing in a circle waiting to be executed.

As the execution begins at some point in the circle, it proceeds around the circle in a fixed direction. During each execution, a specified number of people are skipped, and the next person is executed.

The elimination proceeds around the circle as people are executed, and ends when there is one person remaining, who is given freedom.

Using the double circular linked list, determine the position which will grant freedom.



Assignment Instructions

Complete the functionality for the method `void CDList<T>::getJosephusPosition(int)` inside the `Main.cpp` file.

Once you complete the functionality for the method, ensure that it executes with the supplied `main()` function.

SAMPLE OUTPUT

You can review the output by running the executable in the `OUTPUT` folder.

```
Person 2 was killed.  
Person 4 was killed.  
Person 6 was killed.  
Person 8 was killed.  
Person 3 was killed.  
Person 7 was killed.  
Person 5 was killed.  
Last person left standing is Person 1.
```

NOTE: The `main()` function should not be modified for any reason.

Rubrics

CRITERIA	0 POINTS	1 POINT	2 POINTS	3 POINTS	4 POINTS	5 POINTS
1. LOGIC	Did not complete assigned work	Does not demonstrate ability to use logical process	Poorly demonstrate s ability to use logical process	Somewhat demonstrate s ability to use logical process	Demonstrate s ability to use logical process	Demonstrate s exceptional ability to use logical process
2. EFFICIENCY	Did not complete assigned work	Does not demonstrate any efficiency	Poorly demonstrate s efficiency	Demonstrate s some efficiency	Demonstrate s efficiency	Demonstrate s exceptional efficiency
3. FUNCTIONALITY	Did not complete assigned work	Barely any code is functional and accurate	Parts of code are functional, but are not accurate	Some code is functional and accurate	Most code is functional and accurate	All code is functional and accurate
4. PROCESS & ORGANIZATION	Did not complete assigned work	Very confusing code indentation and/or algorithms	Somewhat confusing code and/or algorithms	Satisfactory code and/or algorithms, but could be improved	Good code and/or algorithms	Excellent code and/or algorithms
5. TEST CASE	Did not complete assigned work	Does not compile	Barely any code functions with test case	Partially functions with test case	Mostly functions with test case	Fully functions with test case
6. TIMELINESS	Did not submit or submitted five, or more days late	Submitted four days late	Submitted three days late	Submitted two days late	Submitted one day late	Submitted on time

Grading standard

30/30 - Work so amazing the instructor would only see this once in a lifetime

25/30 – Exceptional work, rare

20/30 - Great work, student has full command of the topic.

15/30 - Minor errors

10/30 - Errors and perhaps a major error

5/30 - Regular and consistent major errors. Lack of understanding

2/30 - Largely empty