Programming Project – Duck, Duck, Goose

Goal: Write a program that simulates the children's game Duck, Duck, Goose.

Input: A list of children and associated data in a text file, one child per line.

Output: Print the state of the game as it progresses, detailing the results of each step of the game.

Problem Description: In a game of Duck, Duck, Goose, a group of children sit in a circle. One of them starts off as "It" and stands up. The player that is "It" walks around the circle patting each player in the circle on the head, saying "Duck" each time, until randomly identifying another player as "Goose." The Goose stands up and chases the "It" player around the circle. The two children race around the circle to the Goose's former spot, the faster one sitting in it while the player standing becomes "It" for the next round. Between rounds, children occasionally get bored with the game (increases by 5% every round) and go home to play Pokémon. The game ends when fewer than four children remain or an adult calls "Snack Time!" (The adult will call snack time after a fixed number of rounds).

Implement a Circularly-Linked List and use it to simulate children playing the game. Children have names, gender, and age. Older kids run faster, though they do not always run as fast as they can so they are caught by younger kids 50% of the time. Be mindful that your output is grammatically correct and takes into consideration the gender of the children.

Deliverables:

- Your project should be in the package edu.frostburg.cosc310.[LNAME] where LNAME is your last name.
- All source code and any other necessary files will be added to a zip file named [LNAME][FNAME]DuckDuckGoose.zip.
- A Readme.txt file will be included.
- Source code should be documented liberally. (JavaDocs are nice).
- Post-mortem in PDF format.
- All code utilizes OOP best practices and demonstrates professionalism.

Sample output:

```
5 kids play Duck, Duck, Goose.  Michael is it!

Duck, Duck, Duck, Duck, Goose!  Michael tags Tito and he dashes!

Tito runs faster.  Michael is it!

Jermaine leaves to play Pokemon...  4 players remain.

Duck, Duck, Duck, Duck, Duck, Duck, Goose!  Marlon chases Michael!

Michael runs faster.  Marlon is it!

Michael leaves to play Pokemon...  3 players remain.
```

Checklist

Check	Component	20pts
	Program Implementation • Data structures (implementation and use) • Simulation	 10 2
	Professional Code Style	3
	Documentation	3
	Readme.txt	-
	Executable jar file	-
	Post-mortem	2
x	Read specifications	-

Avoid

Check	Component	Grade %
	Compilation errors	-50+%
	Typos	-10+%
	Bugs	-x++
	Other missing components	- y++