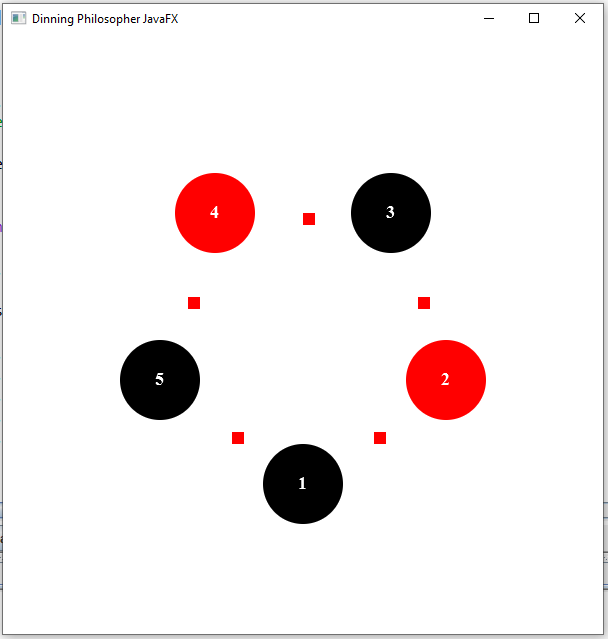
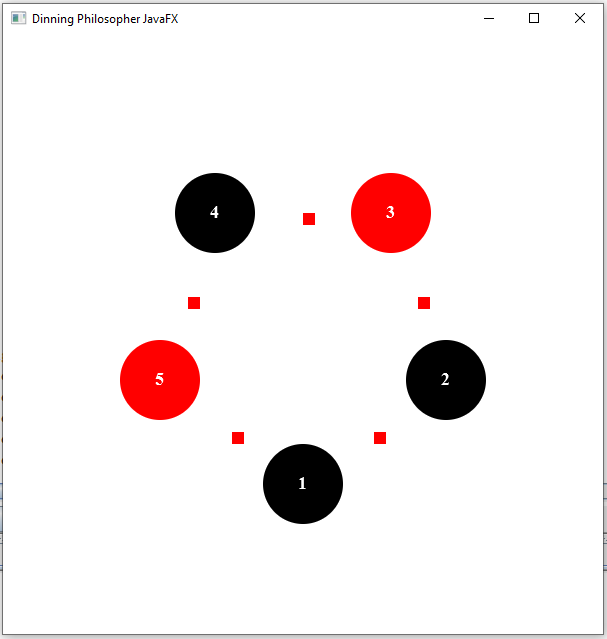
**Programming Assignment 3.1**

**Introduction**

You will revise the java program from programming assignment 3. To visualize the Dining Philosophers problem as shown in the following figures:





**Implementation** (modify only the yellow texts).

**DiningPhilosopherFX.java**

This is the entry point of the program extends from Application.

In this class, the main method must be implemented.

● 5 Chopsticks must be added to an ArrayList of chopsticks

●5 Philosophers must be added to an ArrayList of philosophers

● 5 Philosophers must be initialized (setting the id, and the right and left chopsticks)

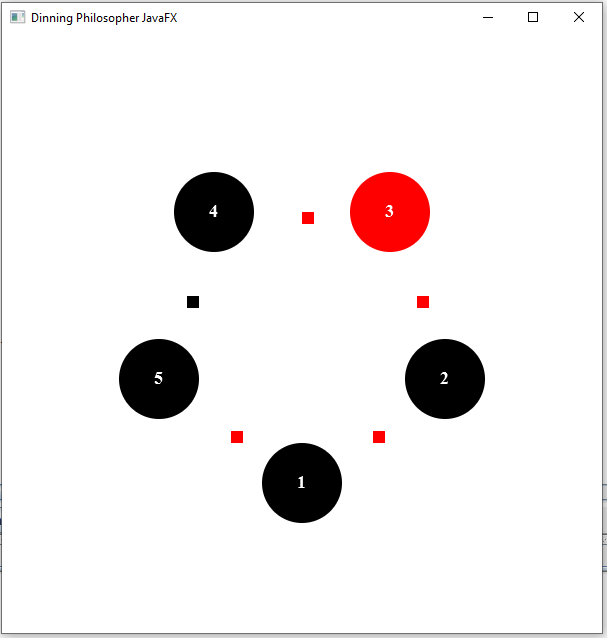
● Then a thread must be created for each philosopher and started.

Make sure to check your code and ensure no deadlocks occur.

Design the Pane layout, add the 5 Circles (Array:ph) represent 5 Philosophers.

And add the 5 Rectangles(Array:ch) display as squares represent 5 Chopsticks.

The program should change the colors of Circles and Rectangles when the Philosopher(s) is/are eating, and the chopstick(s) is/are picked up.



From the above figure, Philosopher 3 can pick up both left and right chopsticks. So, the Philosopher 3 is eating. While Philosopher 2 can pick up only left chopstick and Philosopher 1 can pick up only left chopstick.

**ChopstickFX.java (modify Chopstick.java)**

This class hold information about a Chopstick (whether or not the chopstick is in use, the id, etc.) Must have the following variables:

●id (int) - private

●inUse (bool) - private - indicates whether the chopstick is in use

●Rectangle ch - private

Must have the defined constructor:

●ChopstickFX(int id, Rectangle ch)

* Will set the id , ch and default inUse to false

Will implement the following synchronized methods:

● release() - void

* sets inUse to false
* ~~Out prints to the terminal that the chopstick has been released eg “Chopstick 1has been set down.”~~
* Change the color of Rectangle ch to show the chopstick was released.

● take() - void

* sets inUse to true
* ~~Out prints to the terminal that the chopstick has been released eg “Chopstick 1has been picked up.”~~
* Change the color of Rectangle ch to show the chopstick was taken.

●isInUse() - boolean

* returns inUse

●getId() - int

* returns id

**PhilosopherFX.java (modify Philosopher.java)**

Must have the following variables:

● id (int) - private

● leftChopstick(ChopstickFX) - private

● rightChopstick(ChopstickFX) - private

● Circle ph - private

Must have the defined constructor:

● PhilosopherFX(int id, ChopstickFX leftChopstick, ChopstickFX rightChopstick, Circle ph)

* Will set the id, leftChopstick, rightChopstick, and Circle ph

Will implement the following methods:

● run() - void

* Overrides runnable, initial start into the thread
* Executes eat() method

● eat() - void

* If the left and right chopstick are not in use, pick up chopsticks and
* ~~Outprint “Philosopher X is eating” X being the id of the philosopher.~~
* Change the color of Circle ph to show the philosopher with id is eating.
* Then have the thread sleep for a random time between 0 and 3 secs (i.e.(int)(Math.random()\*3000))
* Release the chopsticks
* Call think()
* This method must handle all exceptions (no throws declaration should be added)

● think() - void

* ~~Outprint “Philosopher X is thinking” X being the id of the philosopher.~~
* Change the color of Circle ph to show the philosopher is thinking.
* Then have the thread sleep for a random time between 0 and 3 secs (i.e.(int)(Math.random()\*3000))
* This method must handle all exceptions (no throws declaration should be added)

**Submission**

Zip up all folders and files under dinner package, the name of a zip file **MUST** be COP3809\_PROGRAM31\_SYY\_XXXX.zip; where YY is 01, 02 or 03 and XXXX is student id.

**Note: I recommend using IntelliJ IDEA, so the program structure, folder and package can be different from the requirement in this assignment.**