

# Project 9: Concurrency and Synchronization

CS 220

## Project objectives

- Use concurrency to speed up a serial program.
- Use synchronization to control concurrent access to shared variables.

## Mandelbrot Set Image Computation

You will parallelize a serial Java program which produces a Mandelbrot set image. Download the Mandelbrot project starter code and import it into Eclipse. Run the program and familiarize yourself with the code, up to and including the `worker` method. (You may ignore the code following the `worker` method. Pay particular attention to the comments.

Refer to the comment block immediately preceding the `worker` method for instructions on how to go about parallelizing this program.

## Pascal Pizza Party Protocol

The CSDS Department is throwing its annual Pascal Pizza Party, with a twist. A pizza maker brings out a single 42 slice pizza at a time, and then goes into a back room to continue devising evil programming projects for his computer architecture students. The students at the party alternate between talking and eating. 40 of the students take a single slice of pizza at a time when they eat. The 41st student takes two slices at a time and the 42nd student takes three slices at a time. If either of these two latter students don't find enough available slices when they want to eat, they wait for the next pizza to appear. The student who takes the last slice of pizza notifies the pizza maker to produce the next pizza.

Write a Java program to solve this problem, using a `ReentrantLock` and `Conditions` to control the concurrency in this problem. The relevant `ReentrantLock` methods are `lock`, `unlock`, and `newCondition`. The Relevant `Condition` methods are `await`, `signal`, and `signalAll`. Refer to the online Java documentation for details on these methods.

Students alternate between talking (A random amount of time between 0 and 1 seconds; use `Thread.sleep` and a per-student `Random` object. Use a unique random seed to initialize each of the students `Random` objects.), and eating (0 delay). The student thread constructor should take as parameters the number of slices to take and a unique id. Students announce (via `println`) how many slices they want when they enter the line, that they are waiting if there aren't enough slices for them currently, the number of slices they took, and the number remaining, and when they request another pie from the pizza maker. The pizza maker announces when he has provided a fresh pie. The pizza maker thread function should take as a parameter the number of slices in a pie. See example output below.

## Project Submission

Export your Eclipse projects individually into two ZIP archives and submit them in Canvas.



Got my 1 slices. 38 remain  
Want 1 slices.  
Got my 1 slices. 37 remain  
Got my 1 slices. 36 remain  
Got my 1 slices. 35 remain  
Got my 1 slices. 34 remain  
Got my 1 slices. 33 remain  
Want 1 slices.  
Got my 1 slices. 32 remain  
Got my 1 slices. 31 remain  
Got my 1 slices. 30 remain  
Want 1 slices.  
Got my 1 slices. 29 remain  
Got my 1 slices. 28 remain  
Want 1 slices.  
Got my 1 slices. 27 remain  
Got my 1 slices. 26 remain  
Got my 1 slices. 25 remain  
Got my 1 slices. 24 remain  
Got my 1 slices. 23 remain  
Want 1 slices.  
Got my 1 slices. 22 remain  
Want 1 slices.  
Got my 1 slices. 21 remain  
Want 1 slices.  
Got my 1 slices. 20 remain  
Want 1 slices.  
Got my 1 slices. 19 remain  
Got my 1 slices. 18 remain  
Got my 1 slices. 17 remain  
Got my 1 slices. 16 remain  
Got my 1 slices. 15 remain  
Want 1 slices.  
Got my 1 slices. 14 remain  
Want 1 slices.  
Got my 1 slices. 13 remain  
Got my 1 slices. 12 remain  
Got my 1 slices. 11 remain  
Got my 1 slices. 10 remain  
Want 1 slices.  
Got my 1 slices. 9 remain  
Want 2 slices.  
Got my 2 slices. 7 remain  
Want 2 slices.  
Got my 2 slices. 5 remain  
Want 3 slices.  
Got my 3 slices. 2 remain  
Want 3 slices.

Gotta wait.  
Want 1 slices.  
Got my 1 slices. 1 remain  
Want 1 slices.  
Got my 1 slices. 0 remain  
Oh, Mr. Pizza Maker...  
Serving a fresh pie.  
Want 2 slices.  
Got my 2 slices. 40 remain  
Want 2 slices.  
Got my 2 slices. 38 remain  
Want 1 slices.  
Got my 1 slices. 37 remain  
Want 1 slices.  
Got my 1 slices. 36 remain  
Want 1 slices.  
Got my 1 slices. 35 remain  
Want 1 slices.  
Got my 1 slices. 34 remain  
Want 1 slices.  
Got my 1 slices. 33 remain  
Want 1 slices.  
Got my 1 slices. 32 remain  
Got my 3 slices. 29 remain  
Want 3 slices.  
Got my 3 slices. 26 remain  
Want 1 slices.  
Got my 1 slices. 25 remain  
Want 1 slices.  
Got my 1 slices. 24 remain  
Want 1 slices.  
Got my 1 slices. 23 remain  
Want 1 slices.  
Got my 1 slices. 22 remain  
Want 1 slices.  
Got my 1 slices. 21 remain  
Want 1 slices.  
Got my 1 slices. 20 remain  
Want 1 slices.  
Got my 1 slices. 19 remain  
Want 1 slices.  
Got my 1 slices. 18 remain  
Want 1 slices.  
Got my 1 slices. 17 remain  
Want 1 slices.  
Got my 1 slices. 16 remain  
Want 1 slices.  
Got my 1 slices. 15 remain

Want 1 slices.  
Got my 1 slices. 14 remain  
Want 1 slices.  
Got my 1 slices. 13 remain  
Want 1 slices.  
Got my 1 slices. 12 remain  
Want 1 slices.  
Got my 1 slices. 11 remain  
Want 1 slices.  
Got my 1 slices. 10 remain  
Want 1 slices.  
Got my 1 slices. 9 remain  
Want 1 slices.  
Got my 1 slices. 8 remain  
Want 1 slices.  
Got my 1 slices. 7 remain  
Want 1 slices.  
Got my 1 slices. 6 remain  
Want 1 slices.  
Got my 1 slices. 5 remain  
Want 1 slices.  
Got my 1 slices. 4 remain  
Want 1 slices.  
Got my 1 slices. 3 remain  
Want 1 slices.  
Got my 1 slices. 2 remain  
Want 1 slices.  
Got my 1 slices. 1 remain  
Want 1 slices.  
Got my 1 slices. 0 remain  
Oh, Mr. Pizza Maker...  
Serving a fresh pie.  
Want 1 slices.  
Got my 1 slices. 41 remain