Homework 10: Multi-Threaded Pi Estimator
CS 412

For this assignment, your job is to implement a multithreaded program that estimates Pi using a Monte Carlo simulation. Your program must take **two** command line arguments: 1) *T*: number of threads and 2) *N*: number of iterations. Your code must then spawn T threads that each run the simulation N times. The code must then print the final estimate for pi.

The Monte Carlo simulation to estimate pi is akin to "throwing darts" at a unit square and counting the number of "darts" that fall within the unit circle. Given the number of hits and attempts, pi is then estimated as:

$$\pi \approx 4\left(\frac{hits}{attempts}\right)$$

REQUIREMENTS:

- Must take two command line arguments:
  - T: number of threads
  - N: number of iterations each thread must run
- Must error check command line options:
  - 0 < T <= 1000
  - 0 < N <= 100000
- Must not result in deadlock
- Must output an estimate of pi
  - More threads and iterations should result in a better estimate
- Must print pi to at least 10 digits
- Must use proper thread synchronization
- Threads must run in parallel
- Main thread must wait for all threads to finish before printing pi estimate

EXAMPLES:

- T = 1; N = 100 => poor estimate
  - *`pi = 3.24000000000000000000`*
- T = 10; N = 100 => better estimate
  - *`pi = 3.19600000000000000000`*
- T = 100; N = 100 => even better estimate
  - *`pi = 3.16600000000000000000`*
- T = 1000; N = 10000 => good estimate
  - *`pi = 3.14226400000000000000`*
- T = 1000; N = 10000 => great estimate
  - *`pi = 3.14148188000000000000`*

HINTS:

- implements Runnable
- Thread
- ArrayList<Thread>
- synchronized method
- double
- .start( )
- .join( )
- main(String[] args)
    - args[0]
    - args[1]
- Math.sqrt(..)
- Math.random( )
- *"if within circle" hits++*
- *Run => Edit Configurations*
    - *Program arguments: 1000 100000*