Overview: In this exercise you will take a serial code for approximately computing pi and add MPI function calls so that all MPI processes collaborate to compute pi.

Monte-Carlo algorithm: randomly choose points in the unit square and at each point check to see if it is in the circle of radius 0.5 centered at x=0.5, y=0.5. Explanation: here.

Serial implementation: we will work together to write a serial implementation.

Source code: the instructor will upload the serial implementation to the course github repo (https://github.com/tcew/CMDA3634FA18) . You should perform the "git pull" operation in your local clone of the course repo. The example code will be in: L12/spi.c

Class Exercise (30 points): make the pi calculator parallel using MPI.

1. (10pts) Follow these instructions to get started:
    a. Make a sub-directory called L12 in your private github repo.
    b. Copy the spi.c source file from the course repo to your L12 sub-directory as mpi.c
    c. Use the git command to add/commit/push the L12 sub-directory to your private github repo.
    d. Check the github website that the sub-directory shows up online.

2. (20pts) In this exercise all the processes will perform the Monte Carlo algorithm, but we will collect all the Ninsides from all ranks on rank 0 to improve the estimate for pi.

    Complete these steps before the end of class:

    a. Add calls to appropriate MPI functions to initialize and finalize the MPI session.
    b. Find the rank of each MPI process.
    c. Find the number (size) of MPI processes.
    d. Use srand48 to set the random number seed to be the MPI process rank.
    e. Add code to:
        i. Make each rank send its value of Ninside to rank 0
        ii. Make rank 0 receive the values of Ninside from all other ranks and combine them to improve its own estimate of pi.
    f. Run with 4 MPI processes, does it converge faster than with one process ?
    g. Raise your hand so we can confirm your findings.
    h. Use the git command to add/commit/push the mpi.c file to your online repo.
    i. Check that your modified mpi.c code appears online.