Katie Bogan 3/15/17

CSC 2300 Calculating Pi

Monte Carlo Pi Method

This project was not entirely difficult. I instantiated my variables and then created the main loop where I did all the math functions with the x and y variables. I then calculated the ratio of the circle to the square, and then multiplied by 4. The law of large numbers tells me that as the number of trials gets larger, my ratio should be close to the expected value of pi, which I what I get. I worked on my own for this project, and I took me about 15 minutes to complete.

**MonteCarlo.java**

import java.util.Random;

import java.math.\*;

public class PiRatio {

public static void main(String[] args) {

int max = 1000000; int count = 0;

double x; double y; double sqTotal;

double distance;

for(int i = 1; i <= max; i++){

x = Math.random();

y = Math.random();

sqTotal = Math.pow(x, 2)+ Math.pow(y, 2);

distance = Math.sqrt(sqTotal);

//System.out.println(distance);

if (distance < 1){

count++;

}

}

double div = (double)count/max;

double ratio = 4.0\*div;

System.out.println("Number of trials: "+max);

System.out.println("Ratio is: "+ratio);

}

}

**Results:**

Number of trials: 1000000

Ratio is: 3.141584

Number of trials: 1000000

Ratio is: 3.142732

Number of trials: 1000000

Ratio is: 3.143216

Number of trials: 1000000

Ratio is: 3.141144

Number of trials: 1000000

Ratio is: 3.141324