**CS419 Project 2: Implementing Disk Scheduling Algorithms**

*You can team up with another student or work alone. Each team can have no more than two students.*

**Objectives:**

1. Reinforce the basic concepts of disk scheduling algorithms.
2. Compare the performance of different disk scheduling algorithms via simulation.

**When it is due:**

See Canvas

**What to submit:**

* Submit only the completed **FCFS.java**, **SCAN.java**, and **CSCAN.java** files.

For a two-person team, you will just need to submit *one* copy (either member submits on Canvas). ***Be sure to include a submission note and a comment header in your code stating it was a team submission with the names of both team members.***

**Instructions:**

1. This project requires you to work on a Java program that simulates disk scheduling using different scheduling algorithms. You will need to read and write multiple Java source files. We highly recommend that you use an IDE, such as [Eclipse](https://www.eclipse.org/) or [IntelliJ IDEA](https://www.jetbrains.com/idea/download/) (the free Community version), to work on this project.
2. You are provided with some source code. You should turn in: FCFS.java, SCAN.java, and CSCAN.java. You are allowed to modify other classes for testing purposes (e.g., to select and run one of your algorithms), but keep in mind that you will only submit these three classes.
3. Your program will service a disk with 5000 cylinders, numbered 0 to 4,999. Each algorithm will be passed the initial position of the disk head (as a parameter) and return an integer representing the total amount of head movement required.
4. Where applicable, assume the disk head always **start** moving from its initial position towards 4,999.
5. Each request is represented by an object with two attributes: *track* and *timeOfArrival*. The former represents which track the head must be positioned at to service the request. The latter represents at what time the request arrives. Note that you cannot service a request before it arrives. If two or more request have the same *timeOfArrival* assume that the order they appear in the list of requests determine which one came first.
6. Assume that 1 unit of space is travelled by the head of the disk in one unit of time. Also assume that servicing a request is instantaneous, once the head is at the right position.
7. To run the project:
   1. Run the *Main* class
   2. Feel free to modify the Main class to test some of your different algorithms, but keep in mind you will only submit FCFS.java, SCAN.java, and CSCAN.java.

**Grading:**

This project carries 60 points. You will receive 20 points for correctly implementing each scheduling algorithms. **Any credit received will be based on the input and output only**. Keep in mind that your algorithms will be tested with input that you do not know.

This project carries 60 points. You will receive 20 points for correctly implementing each scheduling algorithms. **Any credit received will be based on the input and output only**. Keep in mind that your algorithms will be tested with input that you do not know.