# CS 4345 (Operating Systems)

## Assignment – 1 [Spring 2019]

**Due date: 11:00 p.m., Wednesday, February 27, 2019**

Multithreading can help in achieving parallelism in computational problems. This makes the program's response to generate output faster. It is achieved by delegating independent tasks within the program to separate threads instead of creating a sequential routine.

Consider the following sample double array:

| | | | | |
|---|---|---|---|---|
| 3 | 11 | 5 | 19 | 1 |
| 8 | 4 | 16 | 7 | 18 |
| 17 | 6 | 3 | 23 | 9 |

If the problem is to display all the row-sums and all the column-sums, a sequential program would use loops. In that case, the second row (or, column) sum cannot be computed before completing the first row (or, column) sum. However, we can observe that calculating and displaying a particular row (or, column) sum does not depend on calculation and display of other row (or, column) sums. Hence, the inherent parallelism in the problem calls for multithreading.

In this assignment you are required to execute the above task in multithreaded way. In particular, the program should read a double array (as above) from a file (whose name would be supplied by the user) and display all row sums and all column sums. Format of the output should be:

```
Thread i calculates the row-i sum as …
Thread j calculates the column-j sum as …
```

If the double array dimension is *m x n*, then the program will have *m x n* threads to take care of *m* row sums and *n* column-sums. Note, your program may need to decide how many separate threads are needed, after reading the data from the supplied file.

**Submission:**
You can use Java (preferred) to solve the problem. Include your name, course name, semester, and assignment identifier (Assign 1) as program comments at the top of your code. Submit the source file(s) through BlazeVIEW submission box.