CSCI 460: Operating Systems—**Assignment 1 (8 marks)**

This assignment is on processor management. Write a program that will simulate a multi-core non-preemptive process scheduler. The setup is as follows:

(a). Take the last 4 digits of your numeric student number, `std_no`, modulo 3. You need to emulate a multi-core with $k = \texttt{std\_no}\%3 + 2$ processors. For instance, the last 4 digits of your student number is 3456, then you need to emulate a computer with $k = 3456\%3 + 2 = 2$ processors. You can number the processors as 0,1,...,$k-1$.
(b). Write a round-robin procedure to run the job sequence in an on-line fashion, i.e., once a job i arrives you need to, according to the order it arrives, put it on processor $(j+1)\%k$ and run it immediately. Here j is the processor where job $i-1$ was run, and initially $j = 0$, i.e., the first job is run on processor 0. You can assume that it takes 1ms to put each job at any processor to run. When the job sequence is finished, measure the overall *turnaround time*, i.e., the time duration between the arrival of the first job and finish of the last job.
(b.1) Test your program on randomly generated job sequences of 1000 which arrives every 1ms, each job has the processing time between 1ms and 500ms. Run your program at least 100 times to get the statistics, i.e., minimum, maximum, average, and standard deviation.

(b.2) Test your program on the following job sequences of 12. Print the turnaround time. (Again, it takes 1ms to put a job onto any processor.)

Job#	Arrival time	Processing time
1	4	9
2	15	2
3	18	16
4	20	3
5	26	29
6	29	198
7	35	7
8	45	170
9	57	180
10	83	178
11	88	73
12	95	8

(c). Design and code a method which can beat the round-robin method, at least for the test data with 12 jobs (by showing the corresponding turnaround time for the new method). Also, test your program on the sequences of 1000 random jobs, again at least 100 times, then print out the minimum, maximum, average, and standard deviation of the turnaround time.

Date Due: **11:30pm on Monday, October 3, 2016 (on or before 11:30pm, Oct 3, 2016).** Load your source code and output as two separate files on D2L in the folder **Assignment 1**, preferably in the form of **family_name − 1.java** and **family_name − 1.output** (assuming that you are using Java). While discussion is allowed, you MUST finish the assignment by yourself and it does not matter which language you choose to use — of course, it does not hurt to put some comments in the source code on how to run it.