**Programming Assignment 1**



CS450 Fall, 2018

1. This assignment is an individual effort. It is due on 09/12/2018
2. **Requirements:**

This programming assignment is inspired by one given at MIT (<https://pdos.csail.mit.edu/6.828/2017/homework/xv6-shell.html>). The MIT assignment asks the students to implement the pipe “|” and IO redirection “>” operators on the shell program given in https://pdos.csail.mit.edu/6.828/2017/homework/sh.c. We ask you to implement the sequence execution operator “;” and a pair of parenthesis operators in the same shell program. Invoke your shell from the shell that comes with xv6 or from a robust UNIX.

* 1. After your shell has started, it will give a prompt to the user. You should use “CS450$” for the prompt.



* 1. Port a few simple commands such as “echo”, “cat” or “wc” to your environment so that you can test your results. Some of your commands will take arguments.
  2. If the user types cmd1 arg;cmd2 arg2 after the prompt, cmd1 will get executed first. cmd2 will execute after cmd1 terminates. Your shell will give a new prompt in a new line, ready to execute the next command line.
  3. Your shell shall support a command line with a string of 3 or more commands connected by the “;” and parenthesis operators. The operators are processed from left to right. The parenthesis operators change the execution order of the commands; commands enclosed in a pair of parenthesis executes ahead of those outside of the parenthesis. For example, cmd1;(cmd2;cmd3) will see cmd2 executes first, cmd3 executes second and cmd1 executes last.



1. **Deliverables:**
   1. Source and executable objects with a README on how to build and execute them.
   2. A copy of the modified code, showing the modifications that you make and with comments that explains how your code works.
   3. The finite state machine of the parser in UML state chart notations. (e.g. see https://www.ibm.com/support/knowledgecenter/en/SS6RBX\_11.4.2/com.ibm.sa.oomethod.doc/topics/c\_UML\_State\_diag.html)
   4. The test data that you use and the reasons why the test data are of good quality. If you use the equivalence partitioning method to generate your test data, describe your equivalence partitions.

