Eric Altenburg **CS-511** **September 9, 2020**

Mutual Exclusion

1. Atomicity Assumptions on Statements
   - Atomic Operation: An operation is atomic if it cannot be interleaved at a lower level of abstraction
   - For this course: every single-line statement is going to be atomic
     - EX: counter = counter + 1;
   - To simulate something being non-atomic:
     - counter = counter + 1;
       Decomposed to:
       temp = counter + 1;
       counter = temp;
   - A var is a critical ref if:
     1. it is assigned in P and occurs in another process Q, or
     2. it is read in P and is assigned in another process Q
   - Limited Critical Reference (LCR) is satisfied if every statement contains at most one critical reference
   - Concurrent programs that satisfy the LCR restriction yield the same set of behaviors whether the statements are considered atomic or are compiled to a machine architecture with atomic load and store
     - Outputs from writing the transition states and compiling the code will not differ for whether it's atomic or not
2. Race Condition
   - Race Condition arises if two or more threads access the same variables or objects concurrently and at least one does updates
     - EX: order of read and write operations