Goals:
1. Ability to understand and trace pseudo-code to solve a problem (pseudo-code is a structured unambiguous step-by-step procedure to describe how to solve a problem). Examples include HW1 and Quiz 1.

2. Ability to write simple pseudo-code. Examples include:
   i. Solve a linear equation.
   ii. Solve a quadratic equation.
   iii. Multiply two numbers (assuming only addition is known)
   iv. Exponentiate two numbers (assuming multiplication is known)
   v. Sum of first n integers (first n even integers, first n odd integers, sum of squares of first n integers and variations)
   vi. Find the average of n integers
   vii. Sequential search (find an element in a given sequence, find the maximum and minimum element, find all occurrences and locations of a given element and variations)
   viii. Euclid’s GCD (greatest common divisor) Algorithm (from textbook)
   ix. Sort a list of 3 integers in ascending order.
   x. Solve a system of two linear equations in two unknowns.
   Examples include HW 2 and Quiz 2.

3. Truth Tables and Hardware Circuits
   (i) Ability to create truth tables for simple tasks such as majority rule, odd parity circuit typically with up to 3 inputs.
   (ii) Ability to translate the truth table to logical expressions and vice-versa.
   (iii) Ability to translate logical expressions to hardware circuit diagrams and vice-versa.
   Examples include HW 3 and Quiz 3

4. Recursive Algorithms
   (i) Ability to understand and trace simple recursive algorithms;
   (ii) Examples include Factorial Function, Fibonacci Numbers, GCD Algorithm, Hanoi Tower, Binary Search, and Quicksort.
   (iii) Understand and appreciate difference between iterative and recursive algorithms.
   Examples include HW 4