**CISP 360 Midterm Review**

**Midterm Details**

* The midterm will cover chapters 1 through 8.
  + The practice problems below act as a guide but are not an exhaustive list
  + Use the powerpoints, lab solutions, reading, and homework to study
* It will be closed book, closed notes, no computer.
* **The note sheet in Canvas will be printed out and given to you during the exam**

**Practice Problems:**

**Arrays, loops, functions**

**Generate Random Array**

Write a function that takes an array of integers as a parameter, as well as another integer for the size of the array. Create a loop that assigns a random value in the range [0,9] to each element of the array.

**Input Array**

Write a function that takes an array of integers as a parameter, as well as another integer for the size of the array. Create a loop that asks the user to input a number between 0 and 9 for each element of the array. Use an input validation while loop to ensure these numbers are within range. Assign each input to an element of the array.

**Show Array**

Write a function that takes an array of integers as a parameter, as well as another integer for the size of the array. Create a loop that outputs each element of the array.

**Count Matches**

Write a function that takes two array of integers as parameters, as well as another integer for the size of the arrays. Loop through the arrays, count the number of matching digits in the same index, and return the count.

**Main**

Declare two arrays and make them size 5. Ask the user if they would like to play the lottery. Loop as long as their answer is “y”. Within the loop, call your Input Array function. Call the Generate Random Array function. Call the Show Array function and pass the random array into it. Call the Count Matches function and output the count from its return value. If the number if matches is 5, tell them they won the jackpot. Then ask the user if they would like to play again and loop.

**C++ Strings, Streams**

Ask the user to enter a file name. Open the file and read one word at a time. Assume each word is separated by whitespace. Count the number of words that are a palindrome (the same if they are written backwards). Output all palindromes to a file "palindromes.txt". When the file read ends, output the number of palindromes.