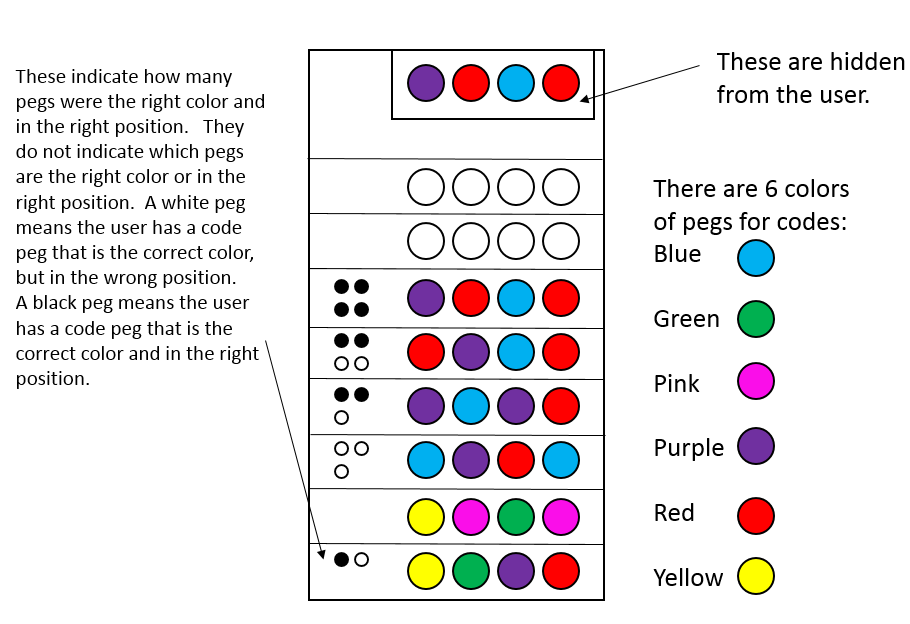
**CSC 117 – Introduction to Computer Science**

**Lab 19 – Mastermind**

This lab has several purposes.

* It will give you practice creating hierarchy charts, flowcharts, and pseudocode to solve a bigger problem than the simple ones at the end of the chapter. Because of the graphics involved, you won’t be expected to do a problem of this magnitude on an exam.
* It will force you to apply most of what you have learned this semester to a single problem. That will require you to think about the tools and techniques you have learned and string several of them together to construct a correct solution.
* It will give you the opportunity to both teach and learn from your classmates. You will be able to see how they attack problems and they will see how you go about solving problems. If you are a strong problem solver, this may force you to learn patience and how to communicate with others and teach them. If you have difficulty solving problems, you will benefit from the experience by understanding what your teammates are doing, and more importantly, WHY they are doing those things. Don’t let them just leap ahead without you! Make sure you understand!

**For this lab, you will write a program to play the game “Mastermind” with the user.**



You have two choices for this lab. For a CHECK, you can implement a text-only version of the program. For a CHECK PLUS, you can implement a version that gets the user input as text, but displays the information to the user graphically. To earn the check or check plus, your program must work correctly. For the text-only version, the letters corresponding to the colors are (B)lue, (G)reen, pin(K), (P)urple, (R)ed, and (Y)ellow. For the small pegs, use (W)hite and (B)lack. The player gets 8 turns to guess the pegs (color and order) that the computer chose. If they get them all right, they win and the computer loses. Otherwise, the computer wins.

Although calculating the number of black pegs isn’t hard, calculating the number of white pegs is tricky, so we’re giving you a way to do that.

The number of white pegs =

(

min(guessed yellow, actual yellow) +

min(guessed red, actual red) +

min(guessed green, actual green) +

min(guessed blue, actual blue) +

min(guessed pink, actual pink) +

min(guessed purple, actual purple)

)

– number of black pegs.

Step 1: Each person should create a hierarchy chart for this program and a list of functions with their arguments and return values that you will need to solve the problem. When you have created your list and feel like you have listed out everything you will need, discuss the chart and list with your partner. When you and your partner have agreed upon a chart and a list, signal the professor that you are ready to continue. When all the teams are ready, we will discuss the lists together as a class. Perhaps another team thought of something extra you don’t need or something else you do need.

Step 2: Now you should plan how to solve the problem. As a team, pseudocode for your program.

Step 3: When you feel like your pseudocode is complete, you may discuss it with your professor if you like and then begin implementing the solution.