Python – May 17 lab

Today, we will practice with 2-dimensional lists and exception handling. Write short Python programs that accomplish the following:

1. Let’s allow the user to play a simple form of Battleship. This is a board game of 8 rows and columns. On this board there are various naval vessels: an aircraft carrier that is 5 units long, a battleship 4 units long, a submarine 3 units long and a destroyer 2 units long.

Initialize the board to be a list of 8 strings, each 8 characters long. Place the ships in the board. For example, your board may start out like this: board = [ “...SSS..”, etc. representing that the submarine is in the first row.

Interactively let the user pick 5 locations to drop bombs. For each bomb, indicate if it is a hit or a miss; if a hit say which vessel has been hit.

1. Let’s discover some Python exceptions. What are the names of the exceptions that occur in the following situations:
	1. Asking for an element beyond the end of a string or list
	2. Converting a string to an int, but the string does not contain digits
	3. Trying to open a file that does not exist.
	4. Dividing by zero.
	5. Taking the square root of a negative number.

Practice writing a try-except block for each case.

1. Write a Python function get\_number that interactively asks a user for a number and returns this value. This function should handle the exception that the user enters text instead. Print an appropriate error message, and let the user try again. You will need to enclose your try-except code inside a loop. You may find this function useful in future programs.

In many situations, we need to input vales, like what you saw with the Goldbach program. Write a main program that asks the user to enter 2 values, with the second at least as large as the first. Call get\_number twice to get 2 input values. Raise an exception if the numbers are out of order, and print an informative error message.

1. The file grades.in contains a set of grades for students taking a class. All the students took three tests, and we need to write a program that can process these scores two-dimensionally. First, read the file and place the numbers into a 2-D list. Then, output the average score for each student, as well as the class average score on each test.

1. Go to lunch. Then work on the following homework assignment due Monday at 9am. Please e-mail your source code to me via e-mail. Modify the Tic-Tac-Toe program so that instead of it being a game between two people at the keyboard, it is a game for a human against the computer.

First, the program should randomly decide who goes first. X will always go first, so the decision is whether X is represented by the person or the machine. The human’s turn will be handled interactively as before. The computer’s turn should be chosen in a pre-determined fashion. There should be an input text file of preferred moves. For example, this input file may contain the values: 5, 1, 3, 7, 9, 2, 4, 6, 8 (one number per line). These numbers indicate that the computer’s first choice is to occupy the middle position. If that is already taken, it will try for the upper left hand corner, the upper right hand corner, and so on. Your program should be written in such a way that it does not know what the preferred list of locations is until it reads the input file. Note that your program will need to “translate” this single number into appropriate row/column values. For example the #5 middle slot is really [1][1] in the board representation.