Final Project
CS4308
Programming Languages

You may choose one of the following three projects as your final project (implemented in C). It is due **Monday, December 7th at 11:59PM**. If you wish to complete the other projects, you may do so for extra credit. For each extra project you correctly implement, you will receive 5 points added to your total Assignment grade (which is 60% of your final grade). Extra credit projects are due by the time of your scheduled final:

Section 1 **(MWF 2:00)** – December 9th3:00
Section 2 **(MWF 11:00)** – December 9th 12:30
Section 3 **(MW 5:00)** – December 9th 6:00

For your submission, you must include the following in a zip folder:

- The .c and makefile (if using Eclipse, you can zip the project and submit that)
- The executable
- A pdf document explaining your approach on how your solved the problem (about half a page) and a screenshot of a test run of your program

Note: Don't reinvent the wheel. If there are standard libraries that can help you, please use them. DO NOT, however, copy code from the Internet. This is to be entirely your work. You may look up syntax online, but the logic should be your own.

Grading:

Requirement	Points
Meets set requirements as stated in problem description (this means that it runs as stated in the problem and directions were followed)	60
PDF write-up of approach (uses appropriate terminology and is a thorough answer)	30
Submission is delivered as requested	10

1. Implement a doubly linked list and the following methods:
   a. Insert
   b. Delete
   c. Add first
   d. Add last
   e. Get
   f. Deep copy
   g. Size

   The main should allow the user to enter three values to insert, a position to delete, a value to add to the beginning of the list, a value to add to the list, a

position to get, and your program should print the resulting list after each input. Finally, the main should print the size of the final list and then make a copy printing out the name, pointer location, and memory location of each list.

2. Create a Hangman game. The following are the requirements for your game.
   a. You should read a list of ten possible words in from a file.
   b. Your game should randomly choose one of the words from the list and display the appropriate number of blanks.
   c. The user should then be able to play the game.
   d. Your game should display the appropriate ASCII art representing the hangman.
   e. If the user chooses to play again, he/she should be able to.
   f. If the user does not choose to play again, you should display the number of wins/losses during the session.

3. Create a calculator. This is a textual calculator not a GUI.
   a. Your calculator should begin by displaying the functions your calculator is able to perform.
   b. The user should then be able to insert a formula for which you calculate and display the answer.
   c. The following functions need to be handled by your calculator:
      i. Sin
      ii. Cos
      iii. Tan
      iv. X^2
      v. X^Y
      vi. Square root
      vii. Addition
      viii. Subtraction
      ix. Multiplication
      x. Division
      xi. Parentheses
   d. The user should be able to enter a formula like:
      2 * (3 + 4) – 21 / 7
      Your calculator should honor the order of operations and give the correct answer of 11.