CSE 1321L: Programming and Problem Solving I Lab
Assignment 4 – 100 points

Iteration

What students will learn:

1) Problem solving
2) Using loops
3) Using logic within loops
4) Nested loops (loops inside of loops)

Overview: If there's one thing that computers do well, it's repeating things. In fact, most of what your computer does is simply "loop" waiting for some kind of input. For this assignment, you're going to use loops to solve some basic problems, and we hope you have fun while you're doing it. The programs aren't long but will require you to think critically. Again, start early, practice, and ask a lot of questions.

Follow the same conventions for class names and file names for your source code. For the Java folks, remove the "package" statement if you have one. Make sure to follow the FYE Submission Guidelines. Finally, we don't mean to lecture, but we want to remind you not to cheat. This is the core of what a lot of you will be doing for a living, so master it now.

Assignment4A: *Diamonds in the sky.*
In our earliest labs, we asked you to print a diamond pattern to the screen using predefined print statements. Now that we know how to use loops, we can make more dynamic and customizable patterns.
For this assignment, we will prompt the user to enter a single character and a maximum width for the diamond. If they enter a number less than 3, we'll prompt them to choose a correct width. If they enter an even number (greater than 3), we will add 1 to it and let the user know the final diamond size. Then we will generate and print out the diamond using the user's inputted character and ' ' symbols.

Hints: Each line of the "diamond" is made up of two parts – the character in the center and the spaces to the left of it. The amount of "left space" decreases as we go towards the middle diamond, then increases afterwards as we go towards the bottom. Could we use multiple loops (or even nested loops) to model this behavior? Also, note that the number of characters increases (and later decreases) by two on each line.

Call the file name Assignment4A(.java, .cs, .cpp) and the class name Assignment4A.
User input is indicated in **bold**

```
Sample Output #1:
Enter a character to use: +
Enter the diamond's width: 5
  +
 +++
+++++
 +++
  +
```

```
Sample Output #2:
Enter a character to use: *
Enter the diamond's width: 8
To make a diamond, we'll use 9 as the width instead.
    *
   ***
  *****
 *******
*********
 *******
  *****
   ***
    *

Sample Output #3:
Enter a character to use: ~
Enter the diamond's width: -2
Please enter a width of at least 3.
Enter the diamond's width: 3
  ~
 ~~~
  ~
```

Assignment 4B: *Mathletes Game!*

With the introduction of loops, we now know enough about programming to begin creating simple games. Our first game (sponsored by the KSU College of Science and Mathematics) will test your ability to solve multiplication problems.

Your game will first ask the player if they want to play on easy or hard mode. Save their response in an appropriate variable (you will reference it later). Then, you will randomly generate two numbers and display them to the player in the form of a multiplication problem (Refer to the Lab 6 instructions and the online lab videos for help with random numbers). The possible numbers will range between -255 and 255.
You will ask the player to solve the equation. You will keep creating multiplication questions until the player wins or loses.

Based on whether the player chose "hard" or "easy" mode, the game will play a little differently. Refer to the following rules for the difference:

Easy Mode	**Hard Mode**
Player only has to answer **3** questions	Player has to answer **6** questions
Player can try again if they get a question wrong (up to two times)	Player loses instantly if they get a question wrong

Call the file name Assignment4B (.java, .cs, .cpp) and the class name Assignment4B. Example outputs are shown below. Note that due to using a random number generator, you will not get the same numerical results – but the formatting and wording should be the same.
User input is indicated in **bold.**

```
Sample Output #1:
[Mathletes Game]
Choose a game mode (0=Easy, 1=Hard): 0
Playing on easy mode, huh? Okay!
Q1. 3 * 4 = ?
12
Correct!
Q2. -3 * 7 = ?
0
Incorrect! Try again.
-21
Correct!
Q3. -1 * -2 = ?
3
Correct!
You win!

Sample Output #2:
[Mathletes Game]
Choose a game mode (0=Easy, 1=Hard): 1
So, you want a challenge? Okay!
Q1. -12 * 7 = ?
-84
Correct!
Q2. -3 * 77 = ?
539
Incorrect!
Game over...
```

Assignment 4C: *Backspacing Animation.*
We covered special characters like \n and \t in a prior module. We also mentioned \b, which replicates the backspace command and "erases" the preceding character when printed to the screen. We're going to use the behavior (along with loops and a new pause command) to make a short animation of a string being erased via backspacing.

For this assignment, you will prompt the user for a sentence. You will also ask them how many letters they want to "erase" with backspacing.
You will print the full sentence to the screen (with no newline character added). Then, you will use a loop to print the \b character the specified number of times. After printing each \b character, you should use also use the sleep command inside the loop to pause the program for half a second. **Information on how to do this is included in the Appendix at the end of the instructions.**

Call the file name Assignment4C (.java, .cs, .cpp) and the class name Assignment4C. Example output is shown below. Note that the last line is the final output – your code's output should gradually remove a letter every half-second until it ends up the same as the final output.
User input is indicated in **bold.**

```
Sample Output:
[Backspace Animation]
Please enter your sentence:
CSE 1321L is my favorite class!
How many letters do you want to backspace?
7
CSE 1321L is my favorite
```

Submission:

1. You will submit 3 separate files, one for each of the assignments above.
2. File names and class names must be correct.
3. Upload both files (simultaneously) to the assignment submission folder in Gradescope.
4. We'll continue to work with you on this assignment if something messes up, so long as you submit by the due date. However, start early because we work during the weekday.

APPENDIX

Making the program "pause" is an advanced topic that will be covered in more detail in CSE 1322. For now, you just need to know how to implement the code in your program. The following code samples will show you how to do this. Note that you will be running these commands more than once – consider what looping structures you've learned in this module that you could use to do this. The relevant parts are highlighted.

Java

```
public class Assignment4C
{
   public static void main(String[] args) throws InterruptedException
   {
      //Pause the program for 0.5 seconds
      Thread.sleep (500);
   }
}
```

C#

```
class Assignment4C
{
   public static void Main(string[] args)
   {
      //Pause the program for 0.5 seconds
      System.Threading.Thread.Sleep (500);

      //Note: You may need to use "\b \b" rather than just "\b"
   }
}
```

C++

```
#include <unistd.h>

int main ()
{
    //Pause the program for 0.5 seconds
    usleep (500000);
    //Makes the changes appear in the console
    cout.flush ();
    //Note: You may need to use "\b \b" rather than just "\b"
}
```