**1050 Programming Logic**

Lab 7 (20 points total)

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Paste your code and screenshots below.*

1. Describe the four basic elements of the counter-controlled repetition.

The four basic elements needed for the counter-controlled repetition are: control variable (often referred to as a loop counter), initial value (starting point of the counter control variable), the increment or decrement (how the variable is changed in the loop), and the loop continuation (where the program execution deciphers if the loop will continue, or end).

1. Compare and contrast the while and for repetition statements.

**The “while” and “for” statement are relatively similar in output. For statements are condensed In less lines horizontally., and while statements is a policy type command list to reference the variables.**

1. Discuss a specific example when it would be more appropriate to use a do-while statement than a while statement. Explain why.

**A do-while statement is more appropriate than a while statement, when the condition at which they run needs to run after the loop. An example of such is:**

**int sum = 0;**

**int i = 0;**

**do**

**{**

**sum += i; i++;**

**}**

**while (i < 100);**

**return sum;**

1. Create a for loop that goes from 1-100 using a variable named i as the counter. Each time through the loop, output whether or not the variable is even or odd

*Hint:* Use and if-else statement and the modulus % operator to determine whether the variable is even or odd. Example: if ((i % 2) == 0) // it’s even

1. Use an if…else-if…else statement to output the following based on an int temp that is input by the user (3 Points) Prompt the user with “Please enter a temperature”.

**Input output**

< 10 Polar Bear

< 20 Penguin

< 40 Moose

< 50 Reindeer

< 60 Deer

< 70 Turtle

< 80 Lion

< 90 Fish

Default Bug

1. The following code is meant to loop and output 10-20, each number on a separate line. What’s wrong?

Fix the problem.

**There isn’t a counter that limits the integer. It runs forever and takes up memory without stopping. It need the “I ++;” command to stop the string. At 20**

int i = 10;

while (i < 21)

{

Console.WriteLine(i);

}

*I ++;*

*}*

*Example output:*



1. *The following statement is supposed to output every number from 0-100 separated by a line with asterisks on it. What is wrong with the code? Fix it.*

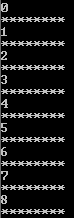
**There aren’t any brackes for the out before the Console and after the “\*\*\*” statement, for the output to be correct. The output references “I” and keeps running**

for (int i = 0; i < 101; i++)

Console.WriteLine(i);

Console.WriteLine("\*\*\*\*\*\*\*\*");

*Example output:*



1. **Extra Credit:** Write an application that displays the following patterns separately, one below the other. Use for loops to generate the patterns. All asterisks (\*) should be displayed by a single statement of the form Console.Write( '\*' ); which causes the asterisks to display side by side. A statement of the form Console.WriteLine(); can be used to move to the next line. A statement of the form Console.Write( ' ' ); can be used to display a space for the last two patterns. There should be no other output statements in the application. [Hint: The last two patterns require that each line begin with an appropriate number of blank spaces.] (2 points possible - +0.5 per solution)

