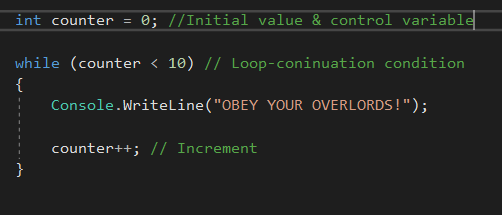
**1050 Programming Logic**

Lab 7 (20 points total)

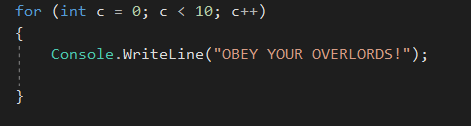
Name: Aaron Osowski

*Paste your code and screenshots below.*

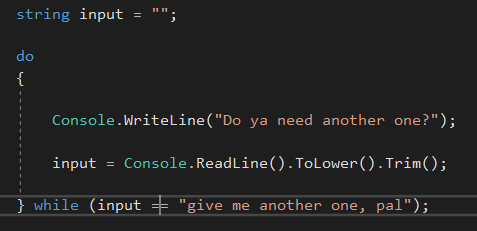
1. *Describe the four basic elements of the counter-controlled repetition.*
   1. A control variable. This is the counter that determines how many times the loop runs
   2. Initial value. This says where the control variable starts. 0, 1, 2, etc…
   3. Increment. This says how much the counter increases per loop. Usually 1.
   4. Loop-continuation condition. This determines if the loop continues or breaks. EX: (while counter <=10)



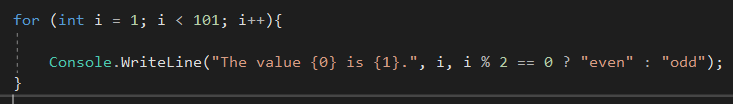
1. *Compare and contrast the while and for repetition statements.*
   1. A while statement needs a counter to be manually declared. For example, in the above screenshot “int counter = 0;”
      1. A for statement has this built in
   2. A while statement must manually increase the counter. For example, in the above screenshot “counter++;”
      1. A for statement has this built in



1. *Discuss a specific example when it would be more appropriate to use a do-while statement than a while statement. Explain why.*
   1. A do while is used when you want the continuation logic checked at the end. This is helpful if you don’t know what the value for your “counter” will be at the beginning. For example, if a user inputs a string and you need to check if that string says “give me another one, pal” before continuing.



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

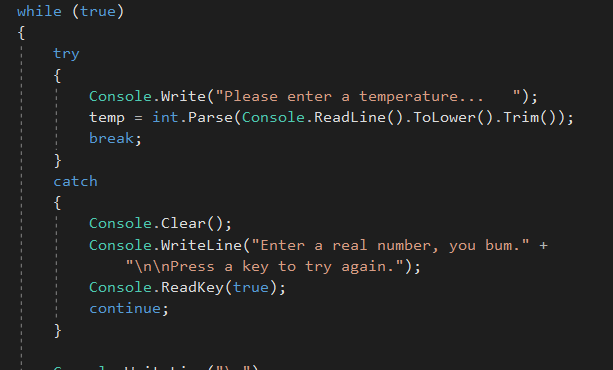
< 60 Deer

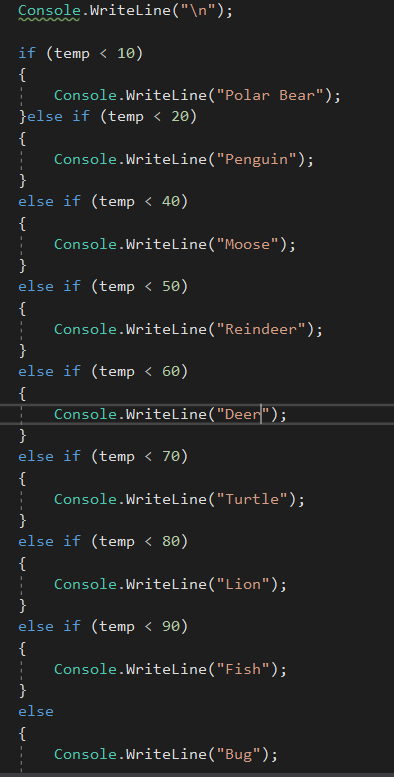
< 70 Turtle

< 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.*

int i = 10;

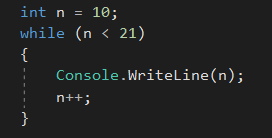
while (i < 21)

{

Console.WriteLine(i);

}

* 1. There is no increment. Add in “i++;”



*Example output:*



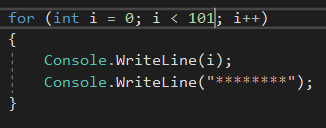
1. *The following statement is supposed to output every number from 0-100 separated by a line with asterisks on it. What is wrong with the code? Fix it.*

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

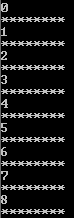
Console.WriteLine(i);

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

* 1. You need to put curly braces around both lines of code.



*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)*

