**1050 Programming Logic**

Lab 5 (20 points total)

Name: **Angela Dennis**

*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, initial value, increment (decrement) and the loop-continuation condition. The control variable is often referred to as a loop counter. The initial value is the starting point of the control variable, increment or decrement is how the control variable is modified each time it loops, and the loop continuation condition is what determines if the script should continue looping or not.

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

The For Loop and While Loop are somewhat of an entry control in that it checks the condition at the beginning. They both repeatedly execute a statement or set of statements as long as a loop-continuation condition remains true. The For Loop is generally right in the header, the While executes both in the beginning and later in the body of the code.

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

If you want some statement or set of statements to execute at least once, then repeat based on a condition, a do... while is more appropriate than a while (or a for). For example if or security reasons you want to check a person's input of information while they are inputting it. Like a password needing to follow a set criteria. Then you can have it checking that the information is up to code WHILE it is being done, rather than tell them after the fact that the password isn't good enough.

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

class Program

{

static void Main(string[] args)

{

for (int i = 1; i < 101; i++)

{

Console.WriteLine(i);

{

if ((i % 2) == 0)

{

Console.WriteLine("Number is even");

}

else if ((i % 2) != 0)

{

Console.WriteLine("Number is odd");

}

}

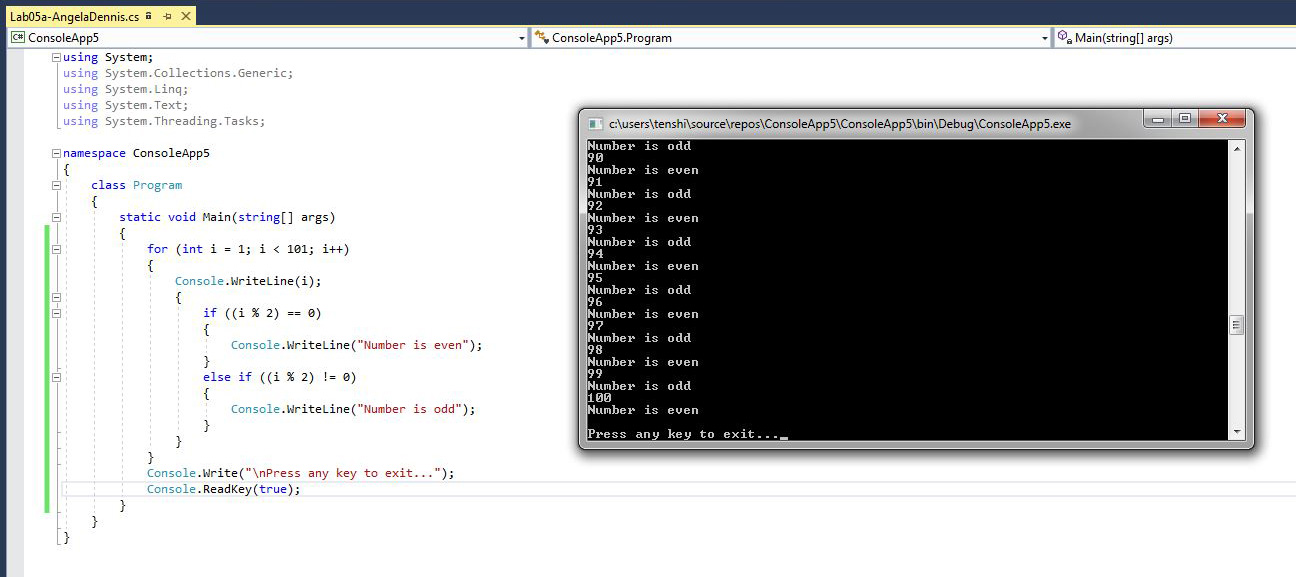
}

Console.Write("\nPress any key to exit...");

Console.ReadKey(true);

}

}



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

Console.Write("Please enter a temperature: ");

int temp = Convert.ToInt32(Console.ReadLine());

if (temp >= 90)

{

Console.WriteLine("Fish");

}

else if (temp >= 80)

{

Console.WriteLine("Lion");

}

else if (temp >= 70)

{

Console.WriteLine("Turtle");

}

else if (temp >= 60)

{

Console.WriteLine("Deer");

}

else if (temp >= 50)

{

Console.WriteLine("Reindeer");

}

else if (temp >= 40)

{

Console.WriteLine("Moose");

}

else if (temp >= 20)

{

Console.WriteLine("Penguin");

}

else if (temp >= 10)

{

Console.WriteLine("Polar Bear");

}

else

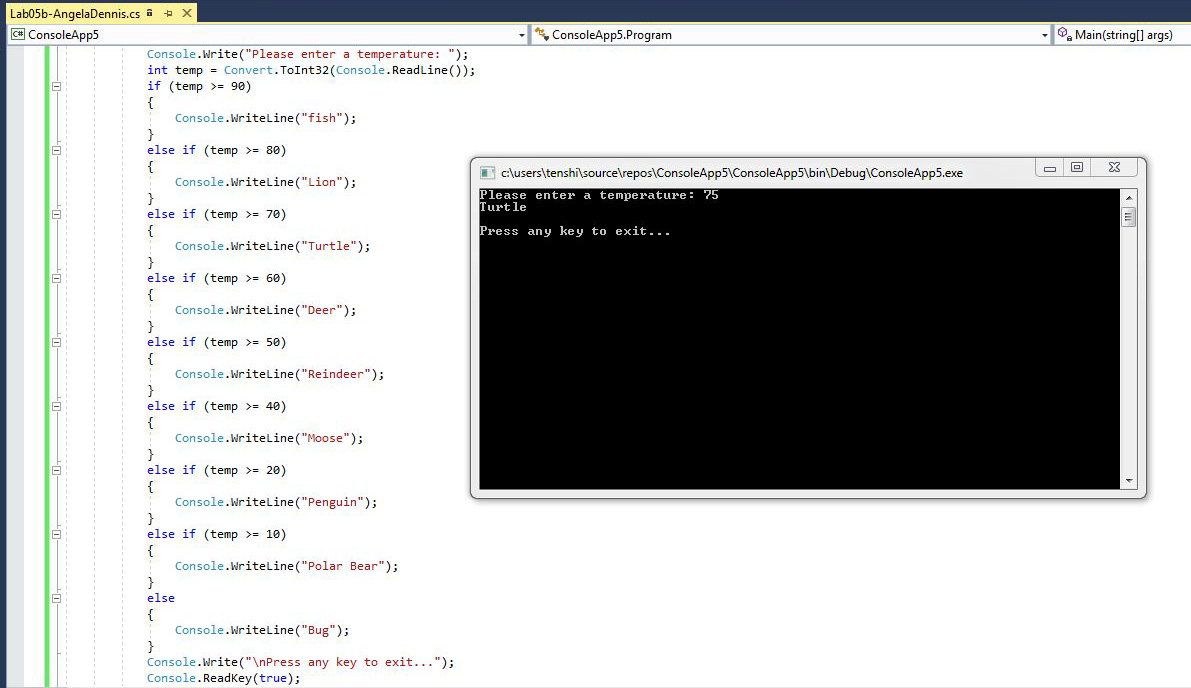
{

Console.WriteLine("Bug");

}

Console.Write("\nPress any key to exit...");

Console.ReadKey(true);



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);

}

*Example output:*



Correct code to display 10-20 is:

static void Main(string[] args)

{

int i = 9;

while (i < 20)

{

i++;

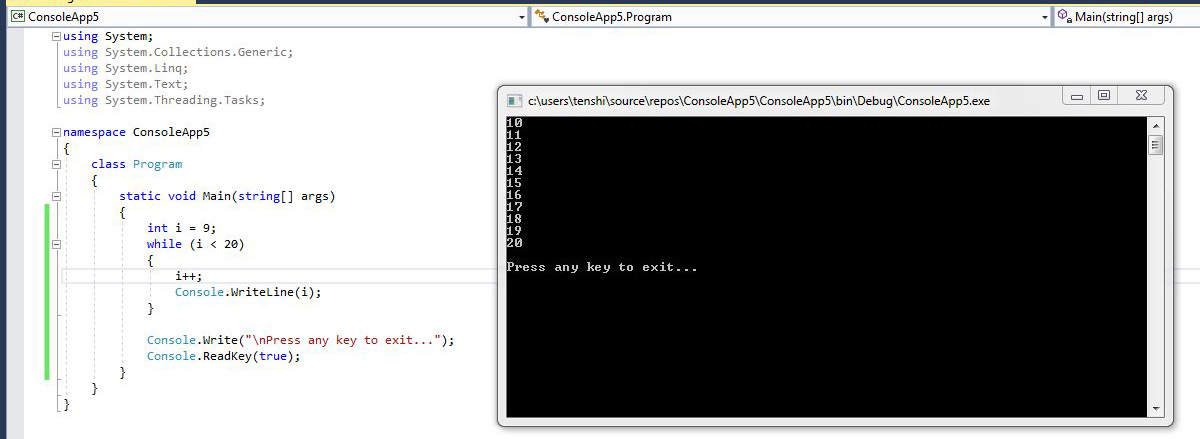
Console.WriteLine(i);

}

Console.Write("\nPress any key to exit...");

Console.ReadKey(true);

}

**

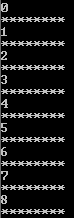
1. *The following statement is supposed to output every number from 0-100 separated by a line with asterisks on it. What is wrong with the code? Fix it.*

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

Console.WriteLine(i);

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

*Example output:*



Correct code to display 0-100 with asterisks is:

static void Main(string[] args)

{

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

{

Console.WriteLine(i);

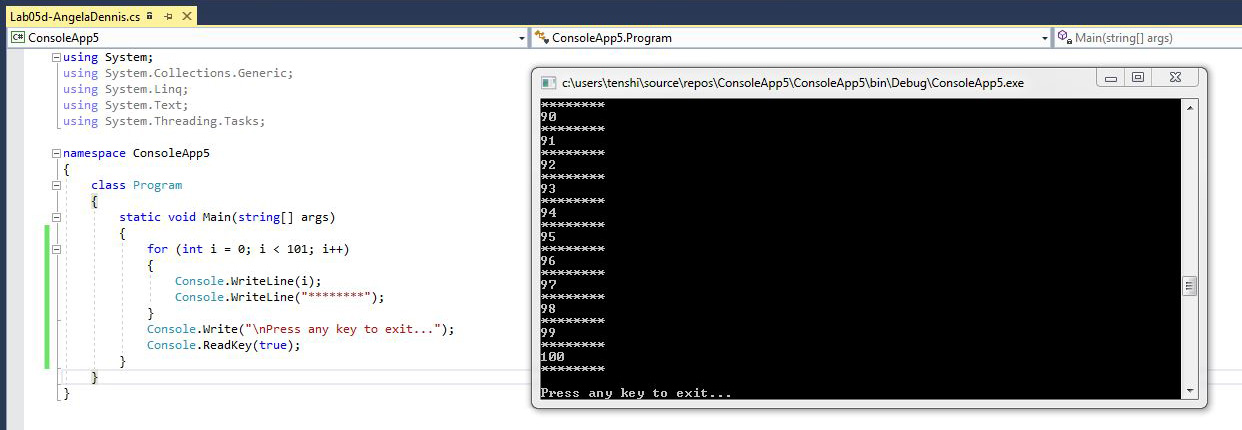
Console.WriteLine("\*\*\*\*\*\*\*\*");

}

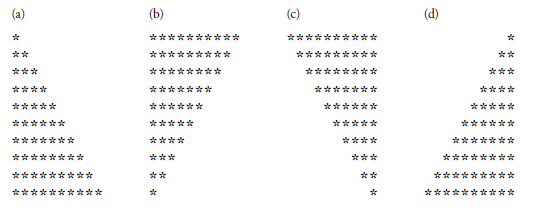
Console.Write("\nPress any key to exit...");

Console.ReadKey(true);

}



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)



static void Main(string[] args)

{

for (int n1 = 1; n1 <= 10; n1++)

{

for (int n2 = 0; n2 < n1; n2++)

Console.Write("\*");

Console.WriteLine();

}

Console.WriteLine();

for (int n1 = 1; n1 <= 10; n1++)

{

for (int n2 = 0; n2 <= 10 - n1; n2++)

Console.Write("\*");

Console.WriteLine();

}

Console.WriteLine();

for (int n1 = 1; n1 <= 10; n1++)

{

for (int n3 = 1; n3 < n1; n3++)

Console.Write(" ");

for (int n2 = 0; n2 <= 10 - n1; n2++)

Console.Write("\*");

Console.WriteLine();

}

Console.WriteLine();

for (int n1 = 1; n1 <= 10; n1++)

{

for (int n3 = 1; n3 <= 10 - n1; n3++)

Console.Write(" ");

for (int n2 = 1; n2 <= n1; n2++)

Console.Write("\*");

Console.WriteLine();

}

Console.Write("\nPress any key to exit...");

Console.ReadKey(true);

}

