Tristan Izlar

COP2360 C# Programming I

**Module 6 ASSIGNMENT Chapter 8 - Problem 10: Morse Code Converter**

SCREENSHOT 1:

Graphical user interface, application

Description automatically generated

SCREENSHOT 2:

CODE:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Morse\_Code\_Converter

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

// exit button

private void exitButton\_Click(object sender, EventArgs e)

{

this.Close();

}

// clear button

private void clearButton\_Click(object sender, EventArgs e)

{

inputTextBox.Text = "";

resultsLabel.Text = "";

}

// convert button

private void convertButton\_Click(object sender, EventArgs e)

{

// creating a 2-D array to house all the possible character-morse code combinations

string[,] translations =

{

{" "," "},

{",","––..––"},

{".",".-.-.-"},

{"?","..––.."},

{"!", "-.-.--"},

{"0","—––––"},

{"1",".—––-"},

{"2","..––-"},

{"3","...––"},

{"4","....-"},

{"5","....."},

{"6","-...."},

{"7","––..."},

{"8","—––.."},

{"9","—-––."},

{"A",".-"},

{"B","-..."},

{"C","-.-."},

{"D","-.."},

{"E","."},

{"F","..-."},

{"G","––."},

{"H","...."},

{"I",".."},

{"J",".––—"},

{"K","-.-"},

{"L",".-.."},

{"M","–––"},

{"N","-."},

{"O","—––"},

{"P",".––."},

{"Q","––.-"},

{"R",".-."},

{"S","..."},

{"T","-"},

{"U","..-"},

{"V","...-"},

{"W",".––"},

{"X","-..-"},

{"Y","-.––"},

{"Z","––.."}

};

// create variables to hold our values

string userInput, morseCode, letterAsStr, dictionaryVal, space;

char letterOfInput;

// space used to separate letters to help with interpretation

space = " ";

// take user input from text box

userInput = inputTextBox.Text;

// turn user input to all uppercase

userInput = userInput.ToUpper();

// start the morseCode string as a blank string

morseCode = "";

// for each character in the user input string

for (int i = 0; i < userInput.Length; i++)

{

// assign letter to variable

letterOfInput = userInput[i];

// convert letter from char type to string

letterAsStr = letterOfInput.ToString();

// for each row of the two-column dictionary array

for (int row = 0; row < 41; row++)

{

// assign a character to a variable

dictionaryVal = translations[row, 0];

// test if the dictionary value is equivalent to the letter of the user input

if (letterAsStr == dictionaryVal)

{

// if true, concatenate the morseCode string with the value that pairs with the dictionary character

morseCode += translations[row, 1] + space;

}

}

}

resultsLabel.Text = morseCode;

}

}

}

// collaboration statement: I worked alone