Tristan Izlar

COP2360 C# Programming I

# **Module 4 TUTORIAL 6-5: Modularizing Input Validation with a Boolean Method**

SCREENSHOT:

Graphical user interface

Description automatically generated

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 Tutorial\_6\_5

{

public partial class Form1 : Form

{

// Constant field for the contribution rate

private const decimal CONTRI\_RATE = .05m;

public Form1()

{

InitializeComponent();

}

// The InputIsValid method converts the user input and stores

// it in the arguments (passed by reference). If the conversion

// is successful, the method returns true. Otherwise it returns

// false

private bool InputIsValid (ref decimal pay,ref decimal bonus)

{

// Flag variable to indicate whether the input is good

bool inputGood = false;

// Try to convert both inputs to decimal.

if (decimal.TryParse(grossPayTextBox.Text, out pay))

{

if (decimal.TryParse(bonusTextBox.Text, out bonus))

{

//Both inputs are good

inputGood = true;

}

else

{

// Display an error message for the bonus.

MessageBox.Show("Bonus amount is invalid.");

}

}

else

{

// Display an error message for gross pay.

MessageBox.Show("Gross pay amount is invalid");

}

// Return the result.

return inputGood;

}

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

{

// Variables for gross pay, bonus, and contributions

decimal grosspay = 0m, bonus = 0m, contributions = 0m;

if (InputIsValid(ref grosspay, ref bonus))

{

// Calculate the amount of contribution.

contributions = (grosspay + bonus) \* CONTRI\_RATE;

// Display the contribution.

contributionLabel.Text = contributions.ToString("c");

}

}

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

{

// Close the form.

this.Close();

}

}

}