Lab 1: On variables and operators

1. Write a console application that prompts for your name and age in sequence and prints out a greeting including your name and age.
2. Modify the MathsOperators project in the \Visual C# Step by Step\Chapter 2\MathsOperators folder to work with double type rather than int type.
3. Solution:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Name\_Age

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter Your Name:");

string name = Console.ReadLine();

Console.WriteLine("Enter Your Age:");

int age = int.Parse(Console.ReadLine());

Console.WriteLine("Hello! I am {0}, and I am {1} years old.", name, age);

Console.WriteLine("Press any key to exit.");

Console.ReadKey();

}

}

}

2. Solution:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace MathsOperators

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void calculateClick(object sender, RoutedEventArgs e)

{

try

{

if ((bool)addition.IsChecked)

addValues();

else if ((bool)subtraction.IsChecked)

subtractValues();

else if ((bool)multiplication.IsChecked)

multiplyValues();

else if ((bool)division.IsChecked)

divideValues();

else if ((bool)remainder.IsChecked)

remainderValues();

}

catch (Exception caught)

{

expression.Text = "";

result.Text = caught.Message;

}

}

private void addValues()

{

double lhs = double.Parse(lhsOperand.Text);

double rhs = double.Parse(rhsOperand.Text);

double outcome;

outcome = lhs + rhs;

expression.Text = lhsOperand.Text + " + " + rhsOperand.Text;

result.Text = outcome.ToString();

}

private void subtractValues()

{

double lhs = double.Parse(lhsOperand.Text);

double rhs = double.Parse(rhsOperand.Text);

double outcome;

outcome = lhs - rhs;

expression.Text = lhsOperand.Text + " - " + rhsOperand.Text;

result.Text = outcome.ToString();

}

private void multiplyValues()

{

double lhs = double.Parse(lhsOperand.Text);

double rhs = double.Parse(rhsOperand.Text);

double outcome;

outcome = lhs \* rhs;

expression.Text = lhsOperand.Text + " \* " + rhsOperand.Text;

result.Text = outcome.ToString();

}

private void divideValues()

{

double lhs = double.Parse(lhsOperand.Text);

double rhs = double.Parse(rhsOperand.Text);

double outcome;

outcome = lhs / rhs;

expression.Text = lhsOperand.Text + " / " + rhsOperand.Text;

result.Text = outcome.ToString();

}

private void remainderValues()

{

double lhs = double.Parse(lhsOperand.Text);

double rhs = double.Parse(rhsOperand.Text);

double outcome;

outcome = lhs % rhs;

expression.Text = lhsOperand.Text + " % " + rhsOperand.Text;

result.Text = outcome.ToString();

}

private void quitClick(object sender, RoutedEventArgs e)

{

this.Close();

}

}

}