**Muzammil Rauf**

**22-Arid-1040**

**Assignment :Advanced programming**

**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 Scientific\_Calculator**

**{**

**public partial class Form1 : Form**

**{**

**public Form1()**

**{**

**InitializeComponent();**

**this.Load += Form1\_Load;**

**}**

**private void Form1\_Load(object sender, EventArgs e)**

**{**

**}**

**private void button33\_Click(object sender, EventArgs e)**

**{**

**string expression = textBox1.Text.Trim();**

**double result;**

**if (TryEvaluateExpression(expression, out result))**

**{**

**textBox1.Text = result.ToString();**

**}**

**else**

**{**

**textBox1.Text = "Error: Invalid expression";**

**}**

**}**

**private bool TryEvaluateExpression(string expression, out double result)**

**{**

**expression = expression.Replace(" ", "");**

**if (expression.StartsWith("(") && expression.EndsWith(")"))**

**{**

**TryEvaluateExpression(expression.Substring(1, expression.Length - 2), out result);**

**}**

**if (expression.Contains("+"))**

**{**

**string[] parts = expression.Split('+');**

**if (parts.Length == 2 && double.TryParse(parts[0], out double num1) && double.TryParse(parts[1], out double num2))**

**{**

**result = num1 + num2;**

**return true;**

**}**

**}**

**else if (expression.Contains("-"))**

**{**

**string[] parts = expression.Split('-');**

**if (parts.Length == 2 && double.TryParse(parts[0], out double num1) && double.TryParse(parts[1], out double num2))**

**{**

**result = num1 - num2;**

**return true;**

**}**

**}**

**else if (expression.Contains("/"))**

**{**

**string[] parts = expression.Split('/');**

**if (parts.Length == 2 && double.TryParse(parts[0], out double num1) && double.TryParse(parts[1], out double num2) && num2 != 0)**

**{**

**result = num1 / num2;**

**return true;**

**}**

**}**

**else if (expression.Contains("\*"))**

**{**

**string[] parts = expression.Split('\*');**

**if (parts.Length == 2 && double.TryParse(parts[0], out double num1) && double.TryParse(parts[1], out double num2))**

**{**

**result = num1 \* num2;**

**return true;**

**}**

**}**

**else if (expression.Contains("%"))**

**{**

**string[] parts = expression.Split('%');**

**if (parts.Length == 2 && double.TryParse(parts[0], out double num1) && double.TryParse(parts[1], out double num2) && num2 != 0)**

**{**

**result = num1 % num2;**

**return true;**

**}**

**}**

**else if (expression.StartsWith("log("))**

**{**

**string numberString = expression.Substring(4, expression.Length - 5);**

**if (double.TryParse(numberString, out double number) && number > 0)**

**{**

**result = Math.Log10(number);**

**return true;**

**}**

**}**

**else if (expression.StartsWith("sin("))**

**{**

**string numberString = expression.Substring(4, expression.Length - 5);**

**if (double.TryParse(numberString, out double number))**

**{**

**result = Math.Sin(number);**

**return true;**

**}**

**}**

**else if (expression.StartsWith("cos("))**

**{**

**string numberString = expression.Substring(4, expression.Length - 5);**

**if (double.TryParse(numberString, out double number))**

**{**

**result = Math.Cos(number);**

**return true;**

**}**

**}**

**else if (expression.StartsWith("tan("))**

**{**

**string numberString = expression.Substring(4, expression.Length - 5);**

**if (double.TryParse(numberString, out double number))**

**{**

**result = Math.Tan(number);**

**return true;**

**}**

**}**

**else if (expression.StartsWith("^"))**

**{**

**string[] parts = expression.Split('^');**

**if (parts.Length == 2 && double.TryParse(parts[0], out double baseValue) && double.TryParse(parts[1], out double exponentValue))**

**{**

**result = Math.Pow(baseValue, exponentValue);**

**return true;**

**}**

**}**

**else if (expression.StartsWith("sqrt("))**

**{**

**string numberString = expression.Substring(5, expression.Length - 6);**

**if (double.TryParse(numberString, out double number) && number >= 0)**

**{**

**result = Math.Sqrt(number);**

**return true;**

**}**

**}**

**else if (expression.ToLower() == "pi")**

**{**

**result = Math.PI;**

**return true;**

**}**

**private void button20\_Click(object sender, EventArgs e)**

**{**

**AppendTextToTextbox("7");**

**}**

**private void button19\_Click(object sender, EventArgs e)**

**{**

**AppendTextToTextbox("8");**

**}**

**private void button18\_Click(object sender, EventArgs e)**

**{**

**AppendTextToTextbox("9");**

**}**

**private void button31\_Click(object sender, EventArgs e)**

**{**

**AppendTextToTextbox("4");**

**}**

**private void button30\_Click(object sender, EventArgs e)**

**{**

**AppendTextToTextbox("5");**

**}**

**private void button29\_Click(object sender, EventArgs e)**

**{**

**AppendTextToTextbox("6");**

**}**

**private void button26\_Click(object sender, EventArgs e)**

**{**

**AppendTextToTextbox("1");**

**}**

**private void button25\_Click(object sender, EventArgs e)**

**{**

**AppendTextToTextbox("2");**

**}**

**private void button24\_Click(object sender, EventArgs e)**

**{**

**AppendTextToTextbox("3");**

**}**

**private void button27\_Click(object sender, EventArgs e)**

**{**

**AppendTextToTextbox("00");**

**}**

**private void button35\_Click(object sender, EventArgs e)**

**{**

**AppendTextToTextbox("0");**

**}**

**private void button34\_Click(object sender, EventArgs e)**

**{**

**if (textBox1.Text == ".")**

**{**

**}**

**else**

**{**

**AppendTextToTextbox(".");**

**}**

**}**

**private void button11\_Click(object sender, EventArgs e)**

**{**

**if (textBox1.Text == "")**

**{**

**}**

**else if (textBox1.Text.Length > 0 && IsOperator(textBox1.Text[textBox1.Text.Length - 1]))**

**{**

**return;**

**}**

**else**

**{**

**AppendTextToTextbox("/");**

**}**

**}**

**private bool IsOperator(char c)**

**{**

**return c == '+' || c=='^' || c == '-' || c == '\*' || c == '/' || c == '^' || c == '%' ;**

**}**

**private void button14\_Click(object sender, EventArgs e)**

**{**

**textBox1.Clear();**

**}**

**private void button13\_Click(object sender, EventArgs e)**

**{**

**if (textBox1.Text.Length > 0)**

**textBox1.Text = textBox1.Text.Substring(0, textBox1.Text.Length - 1);**

**}**

**private void button9\_Click(object sender, EventArgs e)**

**{**

**AppendTextToTextbox("sqrt(");**

**}**

**private void button1\_Click(object sender, EventArgs e)**

**{**

**if (textBox1.Text == "")**

**{**

**}**

**else if (textBox1.Text.Length > 0 && IsOperator(textBox1.Text[textBox1.Text.Length - 1]))**

**{**

**return;**

**}**

**else**

**{**

**AppendTextToTextbox("^");**

**}**

**}**

**private void button4\_Click(object sender, EventArgs e)**

**{**

**AppendTextToTextbox("sin(");**

**}**

**private void button3\_Click(object sender, EventArgs e)**

**{**

**AppendTextToTextbox("cos(");**

**}**

**private void button2\_Click(object sender, EventArgs e)**

**{**

**AppendTextToTextbox("tan(");**

**}**

**private void button7\_Click(object sender, EventArgs e)**

**{**

**AppendTextToTextbox("log(");**

**}**

**private void button17\_Click(object sender, EventArgs e)**

**{**

**if (textBox1.Text == "")**

**{**

**}**

**else if (textBox1.Text.Length > 0 && IsOperator(textBox1.Text[textBox1.Text.Length - 1]))**

**{**

**return;**

**}**

**else**

**{**

**AppendTextToTextbox("\*");**

**}**

**}**

**private void button28\_Click(object sender, EventArgs e)**

**{**

**if (textBox1.Text == "-")**

**{**

**}**

**else if (textBox1.Text.Length > 0 && IsOperator(textBox1.Text[textBox1.Text.Length - 1]))**

**{**

**return;**

**}**

**else**

**{**

**AppendTextToTextbox("-");**

**}**

**}**

**private void button23\_Click(object sender, EventArgs e)**

**{**

**if (textBox1.Text == "+")**

**{**

**}**

**else if (textBox1.Text.Length > 0 && IsOperator(textBox1.Text[textBox1.Text.Length - 1]))**

**{**

**return;**

**}**

**else**

**{**

**AppendTextToTextbox("+");**

**}**

**}**

**private void button5\_Click(object sender, EventArgs e)**

**{**

**if (textBox1.Text == "")**

**{**

**}**

**else if (textBox1.Text.Length > 0 && IsOperator(textBox1.Text[textBox1.Text.Length - 1]))**

**{**

**return;**

**}**

**else**

**{**

**AppendTextToTextbox("^");**

**}**

**}**

**private void button10\_Click(object sender, EventArgs e)**

**{**

**if (textBox1.Text == "")**

**{**

**}**

**else if (textBox1.Text.Length > 0 && IsOperator(textBox1.Text[textBox1.Text.Length - 1]))**

**{**

**return;**

**}**

**else**

**{**

**AppendTextToTextbox("%");**

**}**

**}**

**private void button15\_Click(object sender, EventArgs e)**

**{**

**AppendTextToTextbox("pi");**

**}**

**private void AppendTextToTextbox(string text)**

**{**

**textBox1.Text += text;**

**}**

**private void textBox1\_TextChanged(object sender, EventArgs e)**

**{**

**}**

**}**

**}**