Day_10

Volumn

```csharp
private void button1_Click(object sender, EventArgs e)
        {
            try
            {
                double l = double.Parse(txtl.Text);
                double h = double.Parse(txth.Text);
                double w = double.Parse(txtw.Text);
                double volumn = l * h * w;
        //  txtresult.Text = ;
                txtresult.Text +=" Volumn "+ volumn.ToString() +"cm";
            }
            catch(Exception ex)
            {
                MessageBox.Show(ex.Message);
            }

        }
```

Calculator
```csharp
namespace Calculator
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            label4.Text=   Calculate(int.Parse( textBox1.Text),int.Parse( textBox2.Text),
"+").ToString();

        }

        private double Calculate(int fnamuber, int second, string op)
        {
            double result = 0;
            try
            {
                int istnumber = fnamuber;
                int sndnumber = second;

                if (op == "+")
                {
                    result = istnumber + sndnumber;
                }
                else if(op=="-")
                {
                    result = istnumber - sndnumber;
                }
                else if(op=="*")
                {
                    result = istnumber * sndnumber;
```

```csharp
            }
            else if (op == "/")
            {
                result = istnumber / sndnumber;
            }

          //  label4.Text = result.ToString();
        }
        catch (FormatException fex)
        {
            MessageBox.Show(fex.Message + " Please provide valid number", "Error",
                MessageBoxButtons.OK
                , MessageBoxIcon.Error);
        }
            catch(OverflowException oex)
        {
            MessageBox.Show(oex.Message + " memory overloaded", "Error",
            MessageBoxButtons.OK
            , MessageBoxIcon.Error);

        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);

        }

        finally
        {
            //textBox1.Text = "";
            //textBox2.Text = "";
        }
        return result;
    }

    private void textBox1_TextChanged(object sender, EventArgs e)
    {

    }

    private void button2_Click(object sender, EventArgs e)
    {
        //label4.Text = (int.Parse(textBox1.Text) -
int.Parse(textBox2.Text)).ToString();
       label4.Text=   Calculate(int.Parse(textBox1.Text), int.Parse(textBox2.Text), "-
").ToString();
    }

    private void button3_Click(object sender, EventArgs e)
    {
       label4.Text=   Calculate(int.Parse(textBox1.Text), int.Parse(textBox2.Text),
"*").ToString();
    }

    private void button4_Click(object sender, EventArgs e)
    {
```

```csharp
            label4.Text= Calculate(op:"/",fnamuber:int.Parse(textBox1.Text),second:
int.Parse(textBox2.Text)).ToString();
        }

        private void Form1_Load(object sender, EventArgs e)
        {

        }

    }
}
```

ExceptionHandling

```csharp
    private void button1_Click(object sender, EventArgs e)
        {
            int i = int.MaxValue;
        // MessageBox.Show(i.ToString());
            try
            {
                checked
                {
                    i = i + 1;
                    MessageBox.Show(i.ToString());
                }

            }
            catch(FormatException fex)
            {
                MessageBox.Show(fex.Message);
            }
            catch(OverflowException oex)
            {
                MessageBox.Show(oex.Message);
            }
                catch(InvalidOperationException ioex)
            {

                MessageBox.Show(ioex.Message);
            }
            catch(Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }
```