**C# Other Application Types- Windows Forms, Windows Service**

**Sai Kumaravelu**

**916531**

**Hands on -1**

**Windows service usage**

Business case: Create a Windows service to write a text file with content in a frequent interval

Description: Create a Windows service with a timer that runs every 10 seconds. Configure a time at which an activity needs to be done. The timer should check if the current time is the same or crossed the configured time. If the check is true, then Create a text file and write text ‘Custom activity started at [Current datetime]’

Technical details:

Use System.Timers.Timer, use the elapsed event to do the date time comparison

**Service1.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Diagnostics;

using System.Linq;

using System.ServiceProcess;

using System.Text;

using System.Threading.Tasks;

using System.IO;

using System.Timers;

namespace WindowsService2

{

public partial class Service1 : ServiceBase

{

Timer timer = new Timer();

private static DateTime Scheduled = DateTime.Now.AddSeconds(55);

public Service1()

{

InitializeComponent();

}

protected override void OnStart(string[] args)

{

WriteToFile("Service activity started at " + DateTime.Now);

timer.Elapsed += new ElapsedEventHandler(OnElapsedTime);

timer.Interval = 10000;

timer.Enabled = true;

}

private void OnElapsedTime(object source, ElapsedEventArgs e)

{

if (Scheduled > DateTime.Now)

WriteToFile("Service activity recall at " + DateTime.Now);

}

protected override void OnStop()

{

WriteToFile("Service is stopped at " + DateTime.Now);

}

public void WriteToFile(string Message)

{

string path = AppDomain.CurrentDomain.BaseDirectory + "\\Logs";

if (!Directory.Exists(path))

{

Directory.CreateDirectory(path);

}

string filepath = AppDomain.CurrentDomain.BaseDirectory + "\\Logs\\ServiceLog\_" + DateTime.Now.Date.ToShortDateString().Replace('/', '\_') + ".txt";

if (!File.Exists(filepath))

{

// Create a file to write to.

using (StreamWriter sw = File.CreateText(filepath))

{

sw.WriteLine(Message);

}

}

else

{

using (StreamWriter sw = File.AppendText(filepath))

{

sw.WriteLine(Message);

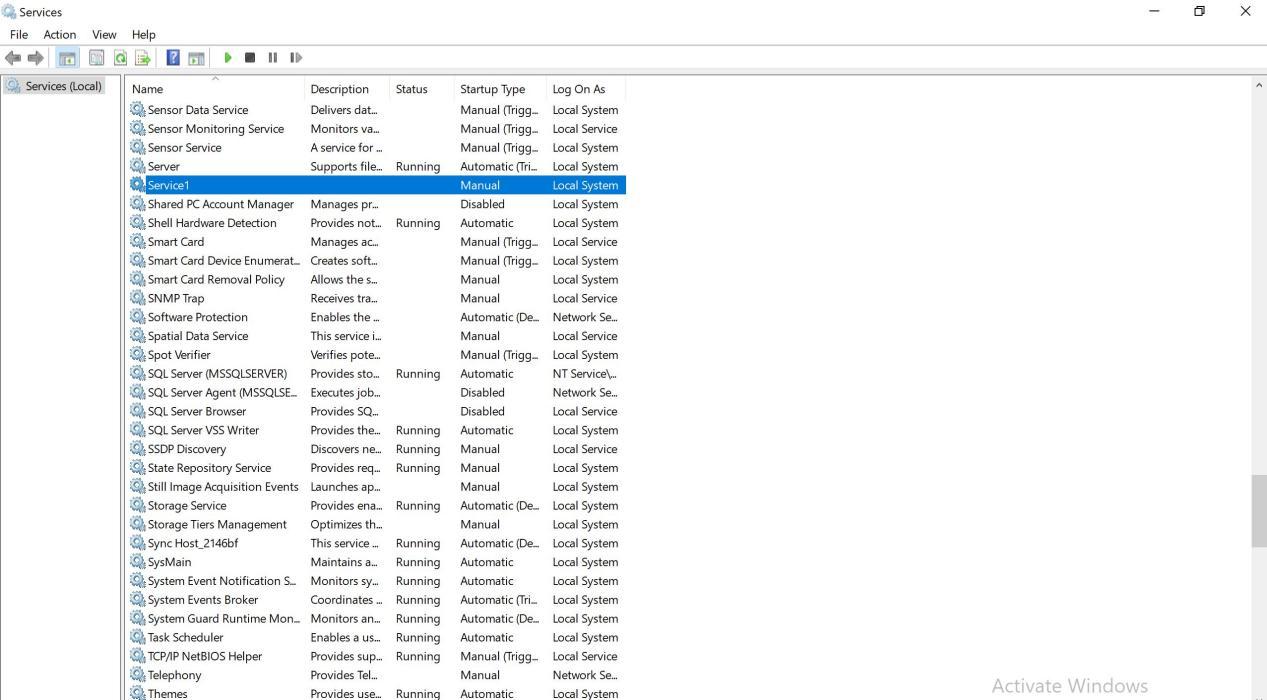
}

}

}

}

}

****

**Windows forms application**

**Handson-2**

Business case: Create a Windows forms application to perform basic Calculator operations

Description: Create a Windows forms application to have

· Two textboxes for input

· Four radio buttons for Addition, Subtraction, Multiplication and Division

· Button to perform the operation

o Check the operation selected from the radio button checked property

· Use MessageBox to show the result

Use regular expression to check the input data. On operation button click, if the input is not a number, then use MessageBox to display message ‘Please enter valid input for the operands’

**Form1.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp3

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void radioButton1\_CheckedChanged(object sender, EventArgs e)

{

}

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

{

string input1 = textBox1.Text;

string input2 = textBox2.Text;

if (!Regex.IsMatch(input1, @"\d+$") || !Regex.IsMatch(input2, @"\d+$"))

{

MessageBox.Show(" Please enter valid input for the operands");

}

if (radioButton1.Checked)

{

MessageBox.Show((int.Parse(input1) + int.Parse(input2)).ToString());

}

else if (radioButton2.Checked)

{

MessageBox.Show((int.Parse(input1) - int.Parse(input2)).ToString());

}

else if (radioButton3.Checked)

{

MessageBox.Show((int.Parse(input1) \* int.Parse(input2)).ToString());

}

else if (radioButton4.Checked)

{

MessageBox.Show((int.Parse(input1) / int.Parse(input2)).ToString());

}

else

{

MessageBox.Show("Please Select a valid Option");

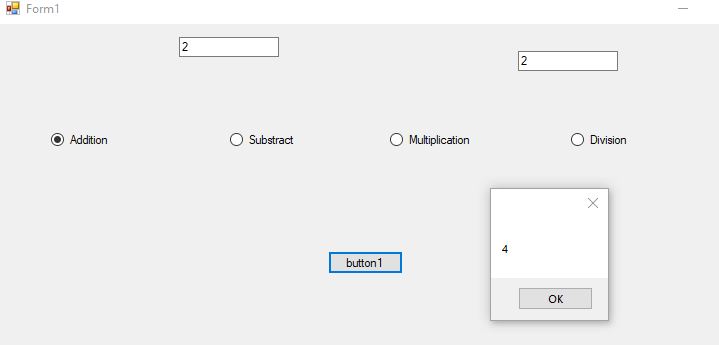
}

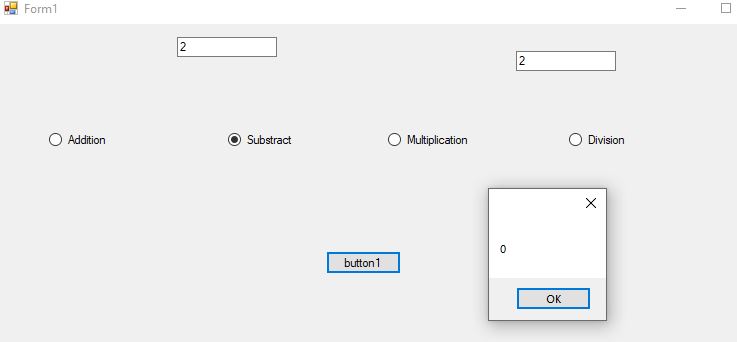
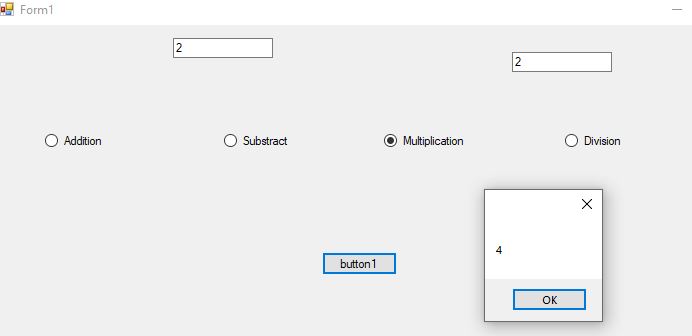
}

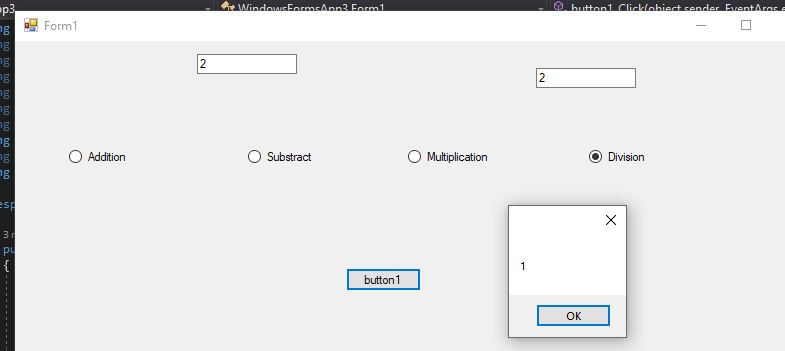
}

}

**Output:**







**Windows forms application**

**Handson-3**

Business case: Create a Windows forms application to provide an option to choose file for upload

Description: Create a Windows forms application to have

· File open dialog to choose an image file. Have restriction to have

· Use PictureBox control to set the image

**Form1.cs**

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 WindowsFormsApp4

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

}

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

{

OpenFileDialog openFileDialog1 = new OpenFileDialog

{

InitialDirectory = @"D:\",

Filter = "image(\*.jpg) | \*.jpg"

};

if (openFileDialog1.ShowDialog() == DialogResult.OK)

{

pictureBox1.SizeMode = PictureBoxSizeMode.StretchImage;

pictureBox1.Image = new Bitmap(openFileDialog1.FileName);

}

}

}

}

**Output:**

