
EXERCISE : 7

Menu Handling in C#

```
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 menuapp
{
    public partial class Form1 : Form
    {
        private ContextMenu popUpMenu;
        private MenuItem Currentcheck;
        private MenuItem checkBlue;
        private MenuItem checkLightBlue;
        private MenuItem checkAqua;
        float currentSize;

        public Form1()
        {
            InitializeComponent();
            popUpMenu = new ContextMenu();
            popUpMenu.MenuItems.Add("Aqua", new EventHandler(popup));
            popUpMenu.MenuItems.Add("Blue", new EventHandler(popup));
            popUpMenu.MenuItems.Add("LightBlue", new EventHandler(popup));
            this.ContextMenu = popUpMenu;
            this.BackColor = Color.Yellow;
            checkAqua = this.ContextMenu.MenuItems[0];
            checkBlue = this.ContextMenu.MenuItems[1];
            checkLightBlue = this.ContextMenu.MenuItems[2];
            Currentcheck = checkAqua;
            Currentcheck.Checked = true;
        }
        private void Form1_Load(object sender, EventArgs e)
        {
            MessageBox.Show("RightClick anywhere On the Form to Get popup Menu");
            textBox1.Text = "This is sample Text, Choose the menu and change the TextBox
properties."+"\r\n"+"If Smaller is selected, get the current size from the Size property
and decrease it. ";
        }
    }
}
```

```

private void popup(object sender, EventArgs e)
{
    Currentcheck.Checked = false;
    MenuItem miClicked = (MenuItem)sender;
    string item = miClicked.Text;
    if (item == "Aqua")
    {
        Currentcheck = checkAqua;
        MessageBox.Show("Background colour will be changed to Aqua");
        this.BackColor = Color.Aqua;
    }
    if (item == "Blue")
    {
        Currentcheck = checkBlue;
        MessageBox.Show("Background colour will be changed to Blue");
        this.BackColor = Color.Blue;
    }
    if (item == "LightBlue")
    {
        Currentcheck = checkLightBlue;
        MessageBox.Show("Background colour will be changed to LightBlue");
        this.BackColor = Color.LightBlue;
    }
    Currentcheck.Checked = true;
    Invalidate();
}

private void centreToolStripMenuItem_Click(object sender, EventArgs e)
{
    textBox1.TextAlign = HorizontalAlignment.Center;
}

private void leftToolStripMenuItem_Click(object sender, EventArgs e)
{
    textBox1.TextAlign = HorizontalAlignment.Left;
}

private void rightToolStripMenuItem_Click(object sender, EventArgs e)
{
    textBox1.TextAlign = HorizontalAlignment.Right;
}

private void biggerToolStripMenuItem_Click(object sender, EventArgs e)
{
    //If Bigger is selected, get the current size from Size property and increase it.
    currentSize = textBox1.Font.Size;
    currentSize += 2.0F;
    textBox1.Font = new Font(textBox1.Font.Name, currentSize,
textBox1.Font.Style, textBox1.Font.Unit);
}

```

```

private void smallerToolStripMenuItem_Click(object sender, EventArgs e)
{
    //If Smaller is selected, get the current size from Size property &decrease it.
    currentSize = textBox1.Font.Size;
    currentSize -= 2.0F;
    textBox1.Font = new Font(textBox1.Font.Name, currentSize,
textBox1.Font.Style, textBox1.Font.Unit);
}

private void exitToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.Close();
}

private void aToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.BackColor = Color.Aqua;
}

private void gToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.BackColor = Color.Blue;
}

private void gToolStripMenuItem1_Click(object sender, EventArgs e)
{
    this.BackColor = Color.Lavender;
}

private void upperCaseToolStripMenuItem_Click(object sender, EventArgs e)
{
    textBox1.CharacterCasing = CharacterCasing.Upper;
}

private void lowerCaseToolStripMenuItem1_Click(object sender, EventArgs e)
{
    textBox1.CharacterCasing = CharacterCasing.Lower;
}

private void boldToolStripMenuItem2_Click(object sender, EventArgs e)
{
    textBox1.Font = new Font(this.Font, FontStyle.Bold);
}

private void italicToolStripMenuItem_Click(object sender, EventArgs e)
{
    textBox1.Font = new Font(this.Font, FontStyle.Italic);
}
}
}
}

```

OUTPUT



