```csharp
//File : \LAB01_nkoirala1\BaseBitmapManip.cs : Last Write 28-Feb-25 8:46:06 AM
using System;
using System.Collections.Generic;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LAB01_nkoirala1
{
    public abstract class BaseBitmapManip
    {
        // constructor requires a source Bitmap, and an error handler
        // the error handler will be used for error notification
        //  when operating the provided bitmap (load, NULL, etc.)
        public BaseBitmapManip(Bitmap bm, Action<string> error)
        {
            try
            {
                // other error handling here?

                // save a copy of the image
                BitmapOriginal = new Bitmap(bm);
            }
            catch (Exception ex)
            {
                error?.Invoke(ex.Message);
            }
        }

        // a COPY of the original bitmap (assigned from the constructor)
        public Bitmap BitmapOriginal { get; private set; }



        // build of a dictionary of colors and their frequencies from
        //  the source image
        // this will tell us the total number of colours in the image
        //  and the results may be ordered to find the most popular
        // colour (something of value for the manipulation)
        public Dictionary<Color, int> BuildColourTable()
        {
            // details omitted - you need to implement this
```

```csharp
            Dictionary<Color, int> result = new Dictionary<Color, int>();

            for (int x = 0; x < BitmapOriginal.Width; x++)
            {
                for (int y = 0; y < BitmapOriginal.Height; y++)
                {
                    Color pixel = BitmapOriginal.GetPixel(x, y);

                    if(result.ContainsKey(pixel))
                        result[pixel]++;
                    else
                        result[pixel] = 1;
                }
            }

            return result;
        }

        // generate a 'difference' value from two colours
        // will be compared against a threshold value
        public static int GetColourDifference(Color A, Color B)
        {
            int iR = Math.Abs(A.R - B.R);
            int iG = Math.Abs(A.G - B.G);
            int iB = Math.Abs(A.B - B.B);

            return iR + iG + iB;
        }

        // abstract method to perform the image manipulation (reduction)
        //  returns a new image that is the reduced version (original unmodified)
        // your derived class will implement this behaviour
        public abstract Bitmap ReduceImage(int Threshold);
    }
}

//File : \LAB01_nkoirala1\Form1.cs : Last Write 28-Feb-25 8:46:06 AM
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
```

```csharp
using System.Linq;
using System.Runtime.InteropServices;
using System.Text;
using System.Threading;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace LAB01_nkoirala1
{
    public partial class Form1 : Form
    {
        private int currentThresVal = 1;
        Action<string> error;
        Bitmap mainImage;
        Bitmap originalImage;
        ImageReducer image;
        private Thread reductionThread;

        public Form1()
        {
            InitializeComponent();

            this.BackColor = Color.LightGray;
            _pictureBOX.BackColor = Color.White;

            _pictureBOX.SizeMode = PictureBoxSizeMode.Zoom;
            _reduceBTN.Enabled = false;

            this.Text = "ImagePress";
            _thresLBL.Text = "Threshold Value:";
            _reduceBTN.Text = "Reduce";
            _resetImage.Text = "Reset Image";
            _consoleLBL.Text = "";
            _thresValueLBL.Text = currentThresVal.ToString();

            error = (msg) => _consoleLBL.Text = msg;

            this.AllowDrop = true;

            this.DragEnter += Form1_DragEnter;
            this.DragDrop += Form1_DragDrop;
            _reduceBTN.Click += _reduceBTN_Click;
            _resetImage.Click += _resetImage_Click;
            _thresValueLBL.MouseWheel += _thresValueLBL_MouseWheel;
```

```csharp
        }

        private void _resetImage_Click(object sender, EventArgs e)
        {
            if (image == null || mainImage == null || originalImage == null) return;

            mainImage = new Bitmap(originalImage);
            _pictureBOX.Image = mainImage;
            image = new ImageReducer(mainImage, error);

            _consoleLBL.Text = "Reseted Image";
        }

        private void _thresValueLBL_MouseWheel(object sender, MouseEventArgs e)
        {
            currentThresVal += e.Delta < 0 ? -1 : 1;

            if (currentThresVal < 1)
                currentThresVal = 1;
            else if (currentThresVal > 256)
                currentThresVal = 256;

            _thresValueLBL.Text = currentThresVal.ToString();
        }

        private void _reduceBTN_Click(object sender, EventArgs e)
        {
            if (image == null || mainImage == null) return;

            _reduceBTN.Enabled = false;
            _pictureBOX.Enabled = false;
            _consoleLBL.Text = "Reducing colors...";

            reductionThread = new Thread(() => ReduceImageThread(currentThresVal));
            reductionThread.Start();
        }
        private void ReduceImageThread(int threshold)
        {
            try
            {
                var stopwatch = System.Diagnostics.Stopwatch.StartNew();

                Bitmap reducedImage = image.ReduceImage(threshold);
```

handwritten annotations: `] background?` `(-5)` `try ? what throws ?`

```csharp
                stopwatch.Stop();
                long elapsedMs = stopwatch.ElapsedMilliseconds;

                Invoke((Action)(() =>
                {
                    _pictureBOX.Image = reducedImage;
                    mainImage = reducedImage;
                    image = new ImageReducer(mainImage, error);
                    _consoleLBL.Text = $"Success - Reduced to {image.BuildColourTable().Count} colors! Took {elapsedMs} ms";
                    _reduceBTN.Enabled = true;
                    _pictureBOX.Enabled = true;
                }));
            }
            catch (Exception ex)
            {
                Invoke((Action)(() =>
                {
                    error($"Error during reduction: {ex.Message}");
                    _reduceBTN.Enabled = true;
                    _pictureBOX.Enabled = true;
                }));
            }
        }

        private void Form1_DragDrop(object sender, DragEventArgs e)
        {
            string[] files = (string[])e.Data.GetData(DataFormats.FileDrop);

            if (files.Length > 0 && System.IO.File.Exists(files[0]))
            {
                try
                {
                    mainImage = new Bitmap(files[0]);
                    _pictureBOX.Image = mainImage;
                    originalImage = new Bitmap(mainImage);

                    image = new ImageReducer(mainImage, error);

                    _consoleLBL.Text = "Success file - " + files[0] + $" - There are {image.BuildColourTable().Count} colors in the image!";

                    _reduceBTN.Enabled = true;
                }
```

```csharp
                    catch
                    {
                        _consoleLBL.Text = "Invalid Image Format";
                        _pictureBOX.Image = null;
                        _reduceBTN.Enabled = false;
                    }
                }
                else
                    _consoleLBL.Text = "File doesn't exist";
        }

        private void Form1_DragEnter(object sender, DragEventArgs e)
        {
            if (e.Data.GetDataPresent(DataFormats.FileDrop))
                e.Effect = DragDropEffects.Copy;
        }
    }
}

//File : \LAB01_nkoirala1\ImageReducer.cs : Last Write 14-Mar-25 2:23:59 PM
using System;
using System.Collections.Generic;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LAB01_nkoirala1
{
    public class ImageReducer : BaseBitmapManip
    {
        public ImageReducer(Bitmap bm, Action<string> error) : base(bm, error)
        {
        }

        public override Bitmap ReduceImage(int Threshold)
        {
            Bitmap result = BitmapOriginal;
            var colorTable = BuildColourTable();

            Dictionary<Color, Color> colorChanged = new Dictionary<Color, Color>();


            while (colorTable.Count > 0)
```

```csharp
264            {
265                List<Color> similarColors = new List<Color>();
266                var popularColor = colorTable.OrderByDescending(kv => kv.Value).First().Key;    → every time?
267
268                foreach (var color in colorTable.Keys.ToList())
269                {
270                    if (GetColourDifference(popularColor, color) <= Threshold)
271                    {
272                        similarColors.Add(color);
273                        colorChanged[color] = popularColor;
274                    }
275                }
276
277                foreach (var color in similarColors)
278                {
279                    colorTable.Remove(color);
280                }
281            }
282
283
284            for (int x = 0; x < result.Width; x++)
285            {
286                for (int y = 0; y < result.Height; y++)
287                {
288                    Color current = result.GetPixel(x, y);
289                    if (colorChanged.ContainsKey(current))
290                        result.SetPixel(x, y, colorChanged[current]);
291                }
292            }
293
294
295            return result;
296        }
297    }
298 }
299
300 //----------------------Git Log----------------------
301 /*commit dca2fd18f7f6d55c60909a4e0f933bb0602cae2d
302 Author: Naresh Koirala <koiralanaresh10@gmail.com>
303 Date:    Sun Mar 2 23:00:25 2025 -0700
304
305     commit
306
307 labs/LAB01_nkoirala1/LAB01_nkoirala1/Form1.Designer.cs
```

Handwritten annotations: "every time?", "remove it & grab new first", "Late. oh, OK"

```
308   labs/LAB01_nkoirala1/LAB01_nkoirala1/ImageReducer.cs
309
310   commit 56cfd3b6d341a9cb6d1237a5ea93e850fc32967d
311   Author: Naresh Koirala <koiralanaresh10@gmail.com>
312   Date:    Tue Feb 25 10:39:08 2025 -0700
313
314       commit
315
316   labs/LAB01_nkoirala1/LAB01_nkoirala1/Form1.cs
317
318   commit 07bce882c89edf182a29b590e8d09ef45c421ca2
319   Author: Naresh Koirala <koiralanaresh10@gmail.com>
320   Date:    Tue Feb 25 03:02:33 2025 -0700
321
322       took me however long to figure the dayum reduce Image i was doing every thing in the while loop which toook to long and then i
          create a dictionary and i was send list of color for similar color took the same time and after 3 cup of coffee at midnight
          figured it out finally
323
324   labs/LAB01_nkoirala1/LAB01_nkoirala1/Form1.Designer.cs
325   labs/LAB01_nkoirala1/LAB01_nkoirala1/Form1.cs
326   labs/LAB01_nkoirala1/LAB01_nkoirala1/ImageReducer.cs
327
328   commit 9e3620d688434692fc407186634ab326610f4ddd
329   Author: Naresh Koirala <koiralanaresh10@gmail.com>
330   Date:    Tue Feb 25 00:47:47 2025 -0700
331
332       reduce click function for the form modified , added a clear picturebox after the drag and drop gets any Invalid file
333
334   labs/LAB01_nkoirala1/LAB01_nkoirala1/Form1.cs
335
336   commit 98092577e98573af2a9e78d2491e73b4fa27fef5
337   Author: Naresh Koirala <koiralanaresh10@gmail.com>
338   Date:    Mon Feb 3 01:23:27 2025 -0700
339
340       done some function - drag drop, mousewheel
341
342   labs/LAB01_nkoirala1/LAB01_nkoirala1/BaseBitmapManip.cs
343   labs/LAB01_nkoirala1/LAB01_nkoirala1/Form1.cs
344   labs/LAB01_nkoirala1/LAB01_nkoirala1/ImageReducer.cs
345
346   commit f9db278d7e24e7e8aeb19ab5730ed993c05b1ba7
347   Author: Naresh Koirala <koiralanaresh10@gmail.com>
348   Date:    Mon Feb 3 00:26:42 2025 -0700
349
```

```
350        made the basebitmapmanip class and also the imagereduce class that is derived from the abtrasct class
351
352    labs/LAB01_nkoirala1/LAB01_nkoirala1.sln
353    labs/LAB01_nkoirala1/LAB01_nkoirala1/App.config
354    labs/LAB01_nkoirala1/LAB01_nkoirala1/BaseBitmapManip.cs
355    labs/LAB01_nkoirala1/LAB01_nkoirala1/Form1.Designer.cs
356    labs/LAB01_nkoirala1/LAB01_nkoirala1/Form1.cs
357    labs/LAB01_nkoirala1/LAB01_nkoirala1/Form1.resx
358    labs/LAB01_nkoirala1/LAB01_nkoirala1/ImageReducer.cs
359    labs/LAB01_nkoirala1/LAB01_nkoirala1/LAB01_nkoirala1.csproj
360    labs/LAB01_nkoirala1/LAB01_nkoirala1/Program.cs
361    labs/LAB01_nkoirala1/LAB01_nkoirala1/Properties/AssemblyInfo.cs
362    labs/LAB01_nkoirala1/LAB01_nkoirala1/Properties/Resources.Designer.cs
363    labs/LAB01_nkoirala1/LAB01_nkoirala1/Properties/Resources.resx
364    labs/LAB01_nkoirala1/LAB01_nkoirala1/Properties/Settings.Designer.cs
365    labs/LAB01_nkoirala1/LAB01_nkoirala1/Properties/Settings.settings
366    */
367
```