

Qualification		Unit number and title	
<b>BTEC Level 3 Extended Diploma in IT</b>		<b>Unit 14: Event Driven Programming</b>	
Student name		Assessor name	
Student Name		Derek Peacock	
Date issued	Hand in deadline	Submitted on	
12 <sup>th</sup> October 2015	22 <sup>nd</sup> November 2015		

Assignment number and title	<b>Assignment 2 : Event Driven Applications</b>
-----------------------------	---

# DIY Room Paint Calculator

By Derek Peacock

## User Requirements Summary

You have been asked to design; implement and test a simple windows desktop application that would assist someone paint a room in their home. The application should prompt the user to enter the dimensions of the room, and it should calculate and display the area of the ceiling, the area of the total area of the four walls and the amount of emulsion pain needed to paint both the walls and ceiling.

Keep this application simple as it is only the first draft of the application, and a more sophisticated version will be developed later. You can leave out any need to correct the areas or paint quantity calculated to take into account any doors and windows in the room. You may if you wish, display how much paint is needed for just the ceiling, and walls separately, as well as the total quantity required if both the walls and ceiling are painted in the same colour.

Please ensure that your final implementation contains sufficient on screen help to ensure that the user is in no doubt as to how to use the calculator. In this first draft no data validation is required, but design the GUI in such a way as to minimise possible mistakes that users could make.

## P4: Calculator Implementation

### Data Input

This is the calculator ready for calculation with three input values already entered.

Room Dimensions		Paint Required	
	Metres	Sq Metres	Litres
Room Length	4.0		
Room Width	3.0		
Room Height	2.0		
		Wall Area	
		Ceiling Area	
		Total Area	

Paint Finish: Matt Emulsion

Requires 2 Coats  
Covers 10 sq metres/litre  
Drying Time 2-4 hours

Buttons: Help, Close, Clear, Calculate

## Output Results

Here is the calculator with the results showing

Derek's Paint calculator

Room Dimensions

	Metres
Room Length	4.0
Room Width	3.0
Room Height	2.0

Paint Required

	Sq Metres	Litres
Wall Area	28	2.8
Ceiling Area	12	1.2
Total Area	40	4.0

Paint Finish

Matt Emulsion

Requires 2 Coats  
Covers 10 sq metres/litre  
Drying Time 2-4 hours

Help Close Clear Calculate

## Clear Form

This shows the form after pressing the clear button

Derek's Paint calculator

Room Dimensions

	Metres
Room Length	
Room Width	
Room Height	

Paint Required

	Sq Metres	Litres
Wall Area		
Ceiling Area		
Total Area		

Paint Finish

Matt Emulsion

Requires 2 Coats  
Covers 10 sq metres/litre  
Drying Time 2-4 hours

Help Close Clear Calculate

## Decorator Form Code

```
namespace DesktopApp
{
    /// <summary>
    /// This class forms the graphics user interface (gui)
    /// for a windows based application where the user can
    /// enter the length, width and height of a room in metres and
    /// the application will calculate the areas of the
    /// ceiling, walls and the combined area in square metres.
    /// The application will also display the quantity of paint
    /// required for the walls, ceiling and both combined in litres.
    ///
    /// Author:          Derek peacock
    /// Date Modified:   12th November 2015
    /// </summary>
    public partial class DecoratorForm : Form
    {
        private PaintCalculator calculator;

        /// <summary>
        /// Initialise the form, clear the form and create
        /// a new instance of the PaintCalculator class
        /// </summary>
        public DecoratorForm()
        {
            InitializeComponent();
            ClearForm(null, null);
            calculator = new PaintCalculator();

            // For Testing Only

            roomHeighttextBox.Text = "2.0";
            roomWidthtextBox.Text = "3.0";
            roomLengthtextBox.Text = "4.0";
        }

        /// <summary>
        /// Change the selected paint finish, i.e the type of paint
        /// to be used. For example silk emulsion or matt emulsion.
        /// Call ShowPaintDetails to display this change.
        /// </summary>
        public void ChangePaint()
        {
            throw new System.Exception("Not implemented");
        }
    }
}
```

```
/// <summary>
/// Clear all the input and output text boxes ready for a
/// new calculation
/// </summary>
private void ClearForm(object sender, EventArgs e)
{
    roomHeighttextBox.Clear();
    roomLengthtextBox.Clear();
    roomWidthtextBox.Clear();

    ceilingAreatextBox.Clear();
    wallAreatextBox.Clear();
    totalAreatextBox.Clear();

    wallPainttextBox.Clear();
    ceilingPainttextBox.Clear();
    totalPainttextBox.Clear();
}

/// <summary>
/// This closes the form and ends the paint calculator
/// </summary>
private void CloseForm(object sender, EventArgs e)
{
    Close();
}

/// <summary>
/// Get the length, width and height of the room in metres,
/// from the textboxes and display an error messages if they
/// are not valid numbers
/// </summary>
public void GetDimensions()
{
    double length;
    double height;
    double width;

    try
    {
        length = Convert.ToDouble(roomLengthtextBox.Text);
        calculator.RoomLength = length;
    }
    catch (Exception)
    {
        MessageBox.Show("This is not a valid length it must be a number!");
    }

    try
    {
        height = Convert.ToDouble(roomLengthtextBox.Text);
        calculator.RoomHeight = height;
    }
    catch (Exception)
    {
        MessageBox.Show("This is not a valid height it must be a number!");
    }

    try
    {

```

```
        width = Convert.ToDouble(roomWidthTextBox.Text);
        calculator.RoomWidth = width;
    }
    catch (Exception)
    {
        MessageBox.Show("This is not a valid width it must be a number!");
    }
}

/// <summary>
/// Use the calculator to calculate the three areas and
/// display the wall area, ceiling area and the total area
/// of the room in square metres
/// </summary>
public void ShowAreas()
{
    calculator.CalculateAreas();

    wallAreaTextBox.Text = calculator.TotalArea.ToString();
    ceilingAreaTextBox.Text = calculator.CeilingArea.ToString();
    totalAreaTextBox.Text = calculator.TotalArea.ToString();
}

/// <summary>
/// Use the calculator to calculate the paint required and
/// then show the quantity of paint in litres for the walls,
/// ceiling and total area of both wall and ceiling
/// </summary>
public void ShowPaint()
{
    calculator.CalculatePaint();

    wallPaintTextBox.Text = calculator.WallPaint.ToString("##.0");
    ceilingPaintTextBox.Text = calculator.CeilingPaint.ToString("##.0");
    totalPaintTextBox.Text = calculator.WallPaint.ToString("##.0");
}

/// <summary>
/// Display the type of paint selected and how many coats
/// are needed, the coverage value and the drying time
/// </summary>
public void ShowPaintDetails()
{
}
}
```

```
/// <summary>
/// Get the dimensions of the room and then call
/// methods to calculate and display the areas of
/// the room and the paint required to paint them.
///
/// This method is called by selecting the calculate
/// button on the decorate form.
/// </summary>
private void CalculatePaint(object sender, EventArgs e)
{
    GetDimensions();

    ShowAreas();

    ShowPaint();
}
}
```

## PaintCalculator Class Code

```
namespace DesktopApp
{
    using System;
    /// <summary>
    /// A class that will use the length, width and height of a room,
    /// and will calculate and return the areas of the room, and how
    /// much paint is needed to paint the wall, ceiling and both
    ///
    /// Author: Derek Peacock
    ///
    /// </summary>
    public class PaintCalculator
    {
        /// <summary>
        /// Room length in metres
        /// </summary>
        private double roomLength;
        public double RoomLength
        {
            set
            {
                roomLength = value;
            }
        }
        /// <summary>
        /// Room width in metres
        /// </summary>
        private double roomWidth;
        public double RoomWidth
        {
            set
            {
                roomWidth = value;
            }
        }
        /// <summary>
        /// Room height in metres
        /// </summary>
        private double roomHeight;
        public double RoomHeight
        {
            set
            {
                roomHeight = value;
            }
        }
        /// <summary>
        /// Area of the four walls in square metres
        /// </summary>
        private double wallArea;
        public double WallArea
        {
            get
            {
                return wallArea;
            }
        }
        /// <summary>
        /// Area of the ceiling in square metres
        /// </summary>
        private double ceilingArea;
    }
}
```

```
public double CeilingArea
{
    get
    {
        return ceilingArea;
    }
}
/// <summary>
/// Combined area of walls and ceiling in square metres
/// </summary>
private double totalArea;
public double TotalArea
{
    get
    {
        return totalArea;
    }
}
/// <summary>
/// Minimum number of coats of paint needed
/// </summary>
private byte numberOfCoats;
public byte NumberOfCoats
{
    get
    {
        return numberOfCoats;
    }
    set
    {
        numberOfCoats = value;
    }
}
/// <summary>
/// The number of square metres covered by one litre of paint
/// </summary>
private double coverage;
public double Coverage
{
    get
    {
        return coverage;
    }
    set
    {
        coverage = value;
    }
}
/// <summary>
/// The volume of paint needed in litres to paint all four walls
/// </summary>
private double wallPaint;
public double WallPaint
{
    get
    {
        return wallPaint;
    }
}
}
```

```
    /// <summary>
    /// The volume of paint needed in litres to paint the ceiling
    /// </summary>
    private double ceilingPaint;
    public double CeilingPaint
    {
        get
        {
            return ceilingPaint;
        }
    }
    /// <summary>
    /// The volume of paint needed in litres to paint all four walls and the
    ceiling combined
    /// </summary>
    private double totalPaint;
    public double TotalPaint
    {
        get
        {
            return totalPaint;
        }
    }

    /// <summary>
    /// Calculate the wall area, ceiling area and total area from the dimensions
    of the room
    /// </summary>
    public void CalculateAreas()
    {
        ceilingArea = roomLength * roomHeight;
        wallArea = 2 * ((roomLength * roomWidth) + (roomWidth * roomHeight));
        totalArea = ceilingArea + wallArea;
    }

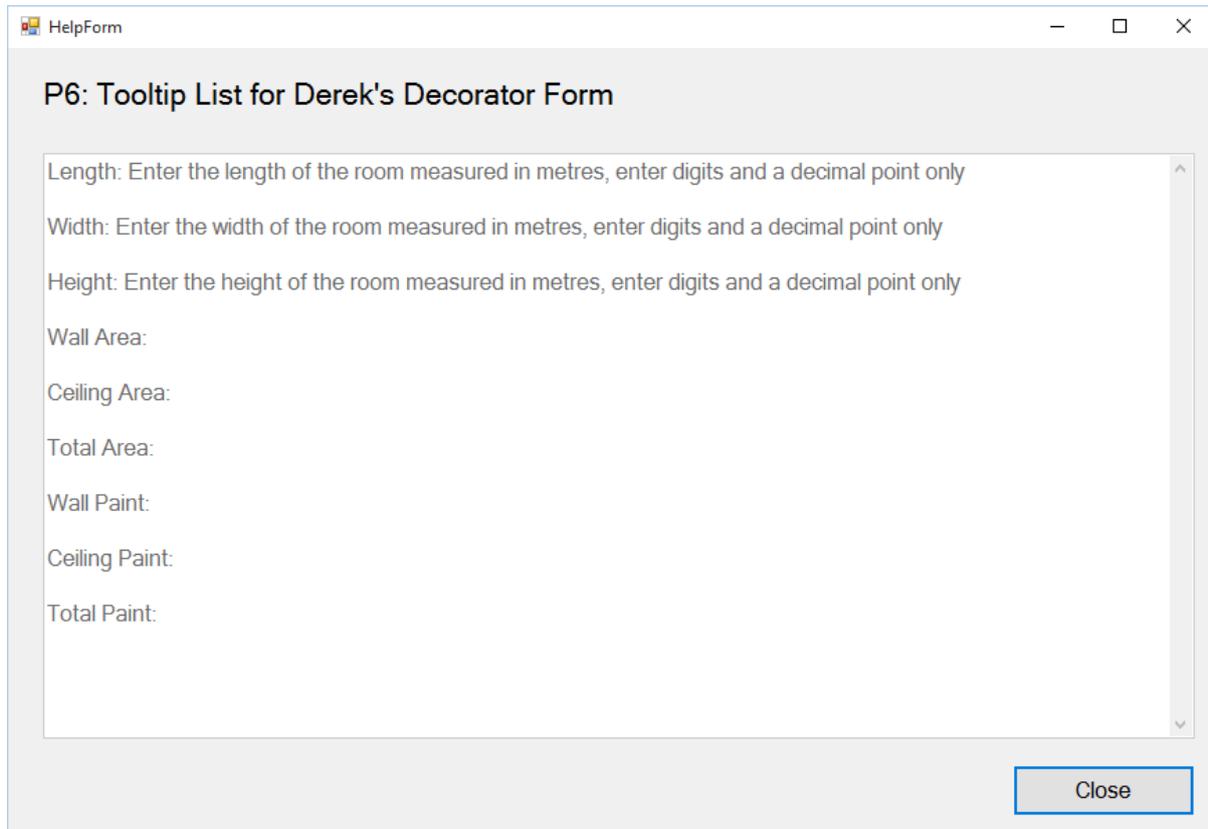
    /// <summary>
    /// Calculate the quantity of paint in litres needed for the wall, the ceiling
    and both combined.
    /// </summary>
    public void CalculatePaint()
    {
        wallPaint = wallArea / coverage;
        ceilingPaint = CeilingArea / coverage;
        totalPaint = totalArea / coverage;
    }
}
}
```

### P5 and M4 Testing the Paint Calculator

Method Tested	Input Data	Expected Results	Actual Results	Comments
Clear Form	Length = 4.0, Width = 3.0, Height = 2.0 Ceiling Area = 12.0 Wall Area = 28.0 Total Area = 40.0 Ceiling Paint = 1.2 Wall Paint = 2.8 Total Paint = 4.0	Length = Width = Height = Ceiling Area = Wall Area = Total Area = Ceiling Paint = Wall Paint = Total Paint =	Length = Width = Height = Ceiling Area = 12.0 Wall Area = Total Area = Ceiling Paint = Wall Paint = Total Paint =	The ceiling Area text box was not cleared because there was no code written to do that in the <b>ClearForm</b> Method.  Code was added for that.
Clear Form	Length = 4.0, Width = 3.0, Height = 2.0 Ceiling Area = 12.0 Wall Area = 28.0 Total Area = 40.0 Ceiling Paint = 1.2 Wall Paint = 2.8 Total Paint = 4.0	Length = Width = Height = Ceiling Area = Wall Area = Total Area = Ceiling Paint = Wall Paint = Total Paint =	Length = Width = Height = Ceiling Area = Wall Area = Total Area = Ceiling Paint = Wall Paint = Total Paint =	Success
Close Form	No data	The form closes	The form closed	Success!
Calculate Areas	Length = 4.0, Width = 3.0, Height = 2.0	Ceiling Area = 12.0 Wall Area = 28.0 Total Area = 40.0	Ceiling Area = 12 Wall Area = 40 Total Area = 40	The wall area is incorrect as instead of multiplying width * height, the calculation multiplied width *length in the <b>CalculateAreas</b> Method. Also there are no decimal places in the output

Method Tested	Input Data	Expected Results	Actual Results	Comments
Calculate Areas	Length = 4.0, Width = 3.0, Height = 2.0	Ceiling Area = 12.0 Wall Area = 28.0 Total Area = 40.0	Ceiling Area = 12.0 Wall Area = 40.0 Total Area = 40.0	The wall area is incorrect still as in the <b>ShowAreas</b> method the <b>totalArea</b> was put into the <b>WallAreaTextBox</b> One decimal place was shown using a suitable format string.
Calculate Areas	Length = 4.0, Width = 3.0, Height = 2.0	Ceiling Area = 12.0 Wall Area = 28.0 Total Area = 40.0	Ceiling Area = 12.0 Wall Area = 28 Total Area = 40.0	Success
Calculate Paint	Length = 4.0, Width = 3.0, Height = 2.0	Ceiling Area = 12.0 Wall Area = 28.0 Total Area = 40.0 Ceiling Paint = 1.2 Wall Paint = 2.8 Total Paint = 4.0	Ceiling Area = 12.0 Wall Area = 28.0 Total Area = 40.0 Ceiling Paint = ∞ Wall Paint = ∞ Total Paint = ∞	The coverage value was not set in the <b>PaintCalculator</b> therefore the areas were divided by zero and anything divided by zero is infinity
Calculate Paint	Length = 4.0, Width = 3.0, Height = 2.0 Coverage = 10 sq metres/ litre	Ceiling Area = 12.0 Wall Area = 28.0 Total Area = 40.0 Ceiling Paint = 1.2 Wall Paint = 2.8 Total Paint = 4.0	Ceiling Area = 12.0 Wall Area = 28.0 Total Area = 40.0 Ceiling Paint = 1.2 Wall Paint = 2.8 Total Paint = 4.0	Success

## P6 Help System



```
private void ShowHelp(object sender, EventArgs e)
{
    HelpForm help = new HelpForm();

    help.HelpText = new string[]
    {
        "Length: " + tooltip1.GetToolTip(roomLengthTextBox),
        " ",
        "Width: " + tooltip1.GetToolTip(roomWidthTextBox),
        " ",
        "Height: " + tooltip1.GetToolTip(roomHeighttextBox),
        " ",
        "Wall Area: " + tooltip1.GetToolTip(wallAreaTextBox),
        " ",
        "Ceiling Area: " + tooltip1.GetToolTip(ceilingAreaTextBox),
        " ",
        "Total Area: " + tooltip1.GetToolTip(totalAreaTextBox),
        " ",
        "Wall Paint: " + tooltip1.GetToolTip(wallAreaTextBox),
        " ",
        "Ceiling Paint: " + tooltip1.GetToolTip(ceilingAreaTextBox),
        " ",
        "Total Paint: " + tooltip1.GetToolTip(totalAreaTextBox),
    };

    help.Show(this);
}
```

## **D2: Evaluate an event driven application**

1. Identify clearly the good points of the current paint calculator application
2. Identify and explain the limitations of the paint calculator application
3. Propose at least 5 ways of improving the application to make it more realistic and more attractive to use.