```
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 Final_Grade_Calculator
{
   public partial class MainForm : Form
   {
      // Declare commonly used variables in public
      int noQuizzes;
      double sumQuizMark;
      double avgQuizMark;

      public MainForm()
      {
         InitializeComponent();
      }

      // When clicking Quit Button, application exits
      private void quitButton_Click(object sender, EventArgs e)
      {
         Application.Exit();
      }
```

```
    // When clicking the arrow(transfer) button,
    private void transferTextButton_Click(object sender, EventArgs e)
    {
        double quizMark;    // Declare the quizMark

        // If user enters non-numeric value in the quizMarkTextbox,
        if (!double.TryParse(quizMarkTextbox.Text, out quizMark))
        {
            // Show the error message below, and allow user to re-enter again
            MessageBox.Show("Missing or invalid non-numeric quiz mark.\nThe value must be
NUMERIC.");
            return;
        }

        // If user enters outside-of-range[0,100] value in the quizMarkTextbox,
        if (quizMark < 0 || quizMark > 100)
        {
            // Show the error message below, and allow user to re-enter again
            MessageBox.Show("Invalid quiz mark.\nThe value entered is outside of range [0,100]");
            return;
        }

        // When user enters the numeric value in between 0 and 100,
        // Show the value entered in multilineQuizMarkTextbox line by line.
        // When the value is transfered, user's cursor will be re-focused again on
        // the empty quizMarkTextbox, to enter the next quiz mark.
        multilineQuizMarkTextbox.AppendText(quizMarkTextbox.Text + Environment.NewLine);
        quizMarkTextbox.Text = "";
```

```
    quizMarkTextbox.Select();
}

// When clicking 'Reset Quiz Marks' button,
private void resetQuizMarksButton_Click(object sender, EventArgs e)
{
    // Make multilineQuizMarkTextbox empty, and reset the sum or
    // average value of quiz marks.
    multilineQuizMarkTextbox.Text = "";
    sumQuizMark = 0;
    avgQuizMark = 0;

    // Focus on quizMarkTextbox
    quizMarkTextbox.Select();
}

// When clicking 'Calculate Grade' button,
private void calculateGradeButton_Click(object sender, EventArgs e)
{
    // Assign the value for the number of Quizzes.
    noQuizzes = multilineQuizMarkTextbox.Lines.Count() - 1;

    // Declare variables
    double max = 0;
    double min = 0;
    double midMark;
    double finalMark;
    double totalMark;
    string letterGrade;
```

```
// Validate if at least one quiz mark is included.
if (noQuizzes < 1)
{
   MessageBox.Show("At least one quiz mark is needed.\nPlease Try again.");
   return;
}

// Validate midterm & finalterm marks;
// For both midterm and finalterm marks, check
// IF user enters no value or non-numeric value, or
// IF user enters outside-of-range[0,100] value.
// In such cases, show the relevant error messages to the user.
if (!double.TryParse(midtermTextbox.Text, out midMark))
{
   MessageBox.Show("Missing or invalid midterm mark.");
   return;
}
if (midMark < 0 || midMark > 100)
{
   MessageBox.Show("Mid-term mark entered is outside of range [0,100]");
   return;
}

if (!double.TryParse(finaltermTextbox.Text, out finalMark))
{
   MessageBox.Show("Missing or invalid finalterm mark.");
   return;
}
```

```
if (finalMark < 0 || finalMark > 100)
{
   MessageBox.Show("Final-term mark entered is outside of range [0,100]");
   return;
}

// Validate calculations on quizzes;
// Allow the user to drop the lowest and the highest quiz marks
// ONLY if the number of quizzes is 5 or more

// In case of dropping the lowest and the highest quiz marks
if (noQuizzes >= 5 && dropCheckbox.Checked == true)
{
   // Get the highest(max) and the lowest(min) quiz marks
   for (int i = 0; i < noQuizzes; i++)
   {
      if (i==0)
      {
         max = double.Parse(multilineQuizMarkTextbox.Lines[i]);
         min = double.Parse(multilineQuizMarkTextbox.Lines[i]);
      }

      else
      {
         if ( double.Parse(multilineQuizMarkTextbox.Lines[i]) > max )
         {
            max = double.Parse(multilineQuizMarkTextbox.Lines[i]);
         }
         if (double.Parse(multilineQuizMarkTextbox.Lines[i]) < min)
```

```
            {
                min = double.Parse(multilineQuizMarkTextbox.Lines[i]);
            }
        }

        // Calculate the sum of total quiz marks,
        // by adding each line of quiz mark in multilineQuizMarkTextbox
        sumQuizMark += double.Parse(multilineQuizMarkTextbox.Lines[i]);
    }

    // Calculate the average of quiz marks;
    // Since the highest(max) and the lowest(min) marks are excluded,
    // the number of quizzes has to be decuted by 2.
    avgQuizMark = (double)(sumQuizMark - max - min) / (noQuizzes - 2);
}

// In case of NOT dropping the lowest and the highest quiz marks
else
{
    // Calculate the sum of total quiz marks,
    // by adding each line of quiz mark in multilineQuizMarkTextbox
    for (int i = 0; i < noQuizzes; i++)
    {
        sumQuizMark += double.Parse(multilineQuizMarkTextbox.Lines[i]);
    }

    // Calculate the average of quiz marks
    avgQuizMark = (double)sumQuizMark / noQuizzes;
}
```

```
// Calculate Total Mark, by weighting the average quiz mark,
// mid-term mark, and the final-term mark differently.
totalMark = 0.2 * avgQuizMark + 0.3 * midMark + 0.5 * finalMark;

// Convert Total Mark to Letter Grade
if (totalMark >= 95 && totalMark <= 100)
{
   letterGrade = "A+";
}
else if (totalMark >= 90 && totalMark < 95)
{
   letterGrade = "A";
}
else if (totalMark >= 85 && totalMark < 90)
{
   letterGrade = "A-";
}
else if (totalMark >= 80 && totalMark < 85)
{
   letterGrade = "B+";
}
else if (totalMark >= 75 && totalMark < 80)
{
   letterGrade = "B";
}
else if (totalMark >= 70 && totalMark < 75)
{
   letterGrade = "B-";
```

```
}
else if (totalMark >= 67 && totalMark < 70)
{
   letterGrade = "C+";
}
else if (totalMark >= 64 && totalMark < 67)
{
   letterGrade = "C";
}
else if (totalMark >= 60 && totalMark < 64)
{
   letterGrade = "C-";
}
else if (totalMark >= 55 && totalMark < 60)
{
   letterGrade = "D+";
}
else if (totalMark >= 50 && totalMark < 55)
{
   letterGrade = "D";
}
else
{
   letterGrade = "F";
}

// Output the Total Mark and Letter Grade
letterGradeTextbox.Text = letterGrade;
percentGradeTextbox.Text = $"{totalMark:F2}".ToString();
```

```
    // In case user wants to quickly modify midterm, finalterm, or quiz marks
    // that they just entered and re-clicks the 'Calculate Grade' button,
    // reset the value, so that new marks can be used.
    midMark = 0;
    finalMark = 0;
    sumQuizMark = 0;
    avgQuizMark = 0;
    noQuizzes = 0;
    totalMark = 0;

}

// When clicking 'Reset All Marks' button,
private void resetMarksButton_Click(object sender, EventArgs e)
{
    // Reset all values
    quizMarkTextbox.Text = "";
    multilineQuizMarkTextbox.Text = "";
    midtermTextbox.Text = "";
    finaltermTextbox.Text = "";
    percentGradeTextbox.Text = "";
    letterGradeTextbox.Text = "";
    sumQuizMark = 0;
    avgQuizMark = 0;
    noQuizzes = 0;
    dropCheckbox.Checked = false;
}
```

```
        // When checking/unchecking 'Drop lowest and highest marks' checkbox,
        private void dropCheckbox_CheckedChanged(object sender, EventArgs e)
        {
            noQuizzes = multilineQuizMarkTextbox.Lines.Count() - 1;

            // IF the number of quizzes is less than 5,
            // Do NOT allow user to 'Drop lowest and highest marks',
            // giving the error message notifying them of why it is not allowed.
            if (noQuizzes < 5)
            {
                MessageBox.Show("This option is available when the numer of quizzes is 5 or more.");
                dropCheckbox.Checked = false;
            }
        }
    }
}
```