A greedy insurance company is looking to automate the process of rating the risk involved in insuring a customer. There are four factors that are known to significantly affect the amount of monies the insurance company may have to pay out if they issue a policy to a customer. Each potential customer starts with rating of 100. A higher rating means the insurance company likes its chances with that customer, and is more likely to issue a policy. A lower rating means the potential customer could wind up costing the insurer huge amounts of money. The risk factors are

(1) Weight: This value is an integer with valid ranges in 0..500 pounds. The insurer subtracts a point for each pound over 250. If no weight information is available, the insurer uses a default value of 350 pounds.

(2) Diet: This is an integer value in pounds of junk food consumed. The insurer subtracts a point for each pound of junk food. The range of junk food consumable is 0 to 100 pounds. If the applicant does not provide information about his or her diet, the insurer assumes he (or she) consumes the maximum amounts of junk food.

(3) Life style: This is an integer that may be a sum of three integers 1, 2, 4, where the integers have the following meaning:

   1 : The applicant regularly exercises: add 20 points.
   2 : The applicant is a couch potato: subtract 30 points.
   4 : The applicant engages in recreational drug use: subtract 60 points.

   For example, a value of 5 means the applicant abuses drugs, but tries to make up for it by exercising regularly. Notice that some values are invalid and should be rejected. For example, a value of 3 is inconsistent because no couch potato exercises regularly. If no information is available, the insurance company assumes the applicant is a drug-addicted couch potato.

   Note that values greater than 7, or less than 1, cannot be valid.

(4) Age: The insurer subtracts a point for each year over 60, but adds a point for each year under 60. In the absence of age information, the insurer assumes an age of 60.

Write a WPF application that displays a labelled textbox for each risk factor. An insurance company employee can enter integer values for all the risk factors. In addition to the risk factors having corresponding text boxes to enter values in, each risk factor is associated with a WPF DependencyProperty with a name such as WeightProperty, LifeStyleProperty, DietProperty, and AgeProperty. There is also an InsuranceRatingProperty that is also a dependency property.

All properties should have default value specified by a PropertyMetadata object whenever appropriate. Also, ValidateValueCallback, and CoerceValueCallback and PropertyChangedCallback delegates should be used whenever possible.

One possible way to proceed is as follows. Create a subclass of Wrapper, say a StackPanel, that represents a Customer for the insurance company. This wrapper will have all the requisite TextBox objects. In addition, you can add the
DependencyProperty properties as needed. Here is a shell that shows only two of the properties, Age and InsuranceRating.

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Shapes;

namespace DependencyPropertyProject
{
    /// <summary>
    /// Interaction logic for Window1.xaml
    /// </summary>
    public partial class Window1 : Window
    {
        public Window1()
        {
            InitializeComponent();
            Content = new Customer();
        }
    }
}

class Customer : StackPanel
{
    TextBox weightTextBox = new TextBox();
    TextBox junkFoodTextBox = new TextBox();
    TextBox lifeStyleTextBox = new TextBox();
    TextBox ageTextBox = new TextBox();
    TextBox ratingTextBox = new TextBox();
    TextBox[] tBoxes;
    String[] descr = {"Weight", "Diet", "LifeStyle", "Age", "Rating"};
    public Customer()
    {
    }

    // Event Handler for when an input text box has a new value
    private void Customer_TextChanged(object sender, RoutedEventArgs e)
    {
    }
}
public static readonly DependencyProperty
    InsuranceRatingProperty;
public static readonly DependencyProperty AgeProperty;

// CLR Wrappers
public int InsuranceRating
{
    set { SetValue(InsuranceRatingProperty, value); } }
    get { return (int)GetValue(InsuranceRatingProperty); }
}

public int Age
{
    set { SetValue(AgeProperty, value); }
    get { return (int)GetValue(AgeProperty); }
}

// static constructor creates property metadata objects for each
// DependencyProperty and then Registers the dependency property
static Customer()
{
    // Metadata for Age
    FrameworkPropertyMetadata pmd = new FrameworkPropertyMetadata(0,
        agePropertyChangedCallback);
    AgeProperty = DependencyProperty.Register("Age", typeof(int),
        typeof(Customer), pmd, ageValidateValueCallback);

    // Metadata for InsuranceRating
    FrameworkPropertyMetadata pmd1 =
        new FrameworkPropertyMetadata(100, InsurancePropertyChangedCallback,
            InsuranceRatingCoerceValueCallback);
    InsuranceRatingProperty = DependencyProperty.Register("InsuranceRating",
        typeof(int), typeof(Customer), pmd1);
}

// ValidateValueCallback for Age
// Checks that age is between 0 and 150 years old
static public Boolean ageValidateValueCallback(Object obj)
{
    MessageBox.Show("Age validate value call back");
}

// PropertyChangedCallback for Age
// Calls CoerceValue on InsuranceRatingProperty
static public void agePropertyChangedCallback(DependencyObject d, DependencyPropertyChangedEventArgs e)
{
    MessageBox.Show("ageProperty changed callback");
}
static public Object InsuranceRatingCoerceValueCallback(DependencyObject d, Object baseValue) {
    MessageBox.Show("insurance property coerce value call back");
}

static public void InsurancePropertyChangedCallback(DependencyObject d, DependencyPropertyChangedEventArgs e) {
    MessageBox.Show("insurance property Changed callback");
}