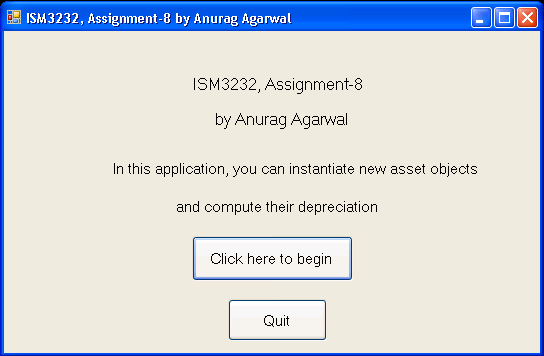
**ISM 3232, Fall-2014, Agarwal**

**Assignment 8, Due Sunday, 11/02/14, 11:59 p.m.**

Create a new Windows application using Visual Basic .NET 2010 and name it ISM3232Assign8LastNameFirstName. This will be a multi-form application involving a user-defined class. It will also involve exception handling. The application will contain two forms and a class.

The first form, the opening form looks something like this:

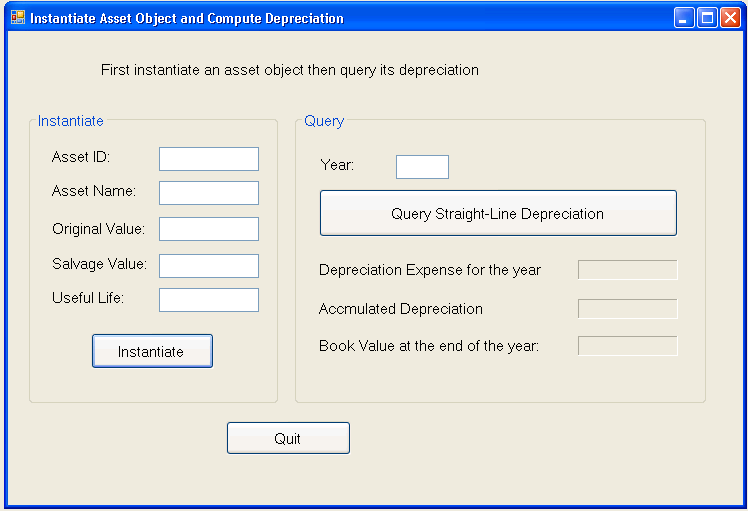


You will of course use your name instead of mine on the form. The Quit button should, of course, end the application. The “Click here to begin” button should open the second form.

You will create a user-defined class with an appropriate name for asset objects. This class will have five properties and one method in the form of a sub. The five properties are for asset id, asset name, original value, salvage value and useful life. The method is to calculate the depreciation, accumulated depreciation and book value using the straight-line depreciation method. Asset id and asset name are string properties, original and salvage value properties should be able to handle decimal values. Useful life can only have integer values. The method to calculate depreciation should take one input parameter called “year” and should have three output parameters – one for depreciation expense for the given year, one for the accumulated depreciation at the end of the year and one for the book value at the end of the year.

On the second form, which you will name appropriately, there are two group boxes and a quit button. The quit button should close the form. The first group box has five sets of labels and textboxes for the user to enter the five attributes of an asset object. The instantiate button should perform extensive exception handling before assigning the values in the five text boxes to the corresponding properties. The second group box is to perform the query for depreciation for a given year that the user enters in a text box.

The second form should look something like this:



Exceptions to Throw by the Instantiate button:

1. Asset ID should not be blank
2. Asset Name should not be blank
3. Original value should not be blank, should be numeric, should be positive and should not exceed 1,000,000
4. Salvage value should not be blank, should be numeric, should be positive and should not exceed 1,000,000, and should not be greater than the original value.
5. Useful life should not be blank, should be numeric, positive and should not exceed 50.

If none of the above 15 exceptions are thrown, the asset object should be instantiated and all values from the text boxes should be assigned to the corresponding properties of the object and a message should be displayed that says that the asset object has been instantiated and that the user can proceed with the query.

When the “Query” button is pressed, the following should happen:

1. It should be validated that the user has first created an object. This can be done by checking if the asset id is blank or not. If asset id is blank, there should be an exception thrown with a message that the user should first instantiate an asset object before trying to query it.
2. The year field should be validated to ensure it is not blank, is numeric, is positive and is not greater than the life of the asset for that object.
3. The method should be called by passing the year value as the input parameter and variables for three output values (depreciation, accumulated depreciation and book value).
4. The depreciation value for the year, the accumulated depreciation at the end of the year and the book value at end of that year should be displayed in the appropriate labels using the currency format.

Zip up the application as ISM3232Assgn8LastNameFirstName.zip and submit it through Canvas by the deadline.