Assignment 5 - Visual Basic - Introductory Program, Basic Controls, Properties, Assignment, Expressions, Selection

Assignment Specifications

Overview

You are to develop a leasing assistant app for the Haleakela Lofts development, a two-building complex of loft housing located on Maui in the state of Hawaii next to the beach. The complex has two buildings: **Crashing Surf** and **SunView**. The Crashing Surf building has direct whitewater views of the beach, and is more expensive in which to rent. The leasing assistant app allows the user to enter a base rent (which would be the rent charged for a standard [default] loft) and then to select a variety of attributes about a loft:

* location (building)
* lanai or not
* rooftop deck or not
* non-smoking unit or not

The combination of attributes determines the ultimate monthly rent to be paid. Actual rent may be lower than Base Rent due to discounts provided by certain attributes (in this program's case, just the non-smoking unit).

The attributes are to be selected, and then upon pressing of an on-screen Calculate Rent button, the rent is calculated and displayed.

You are to develop this program using Visual Basic.

User Interface and Functional Specifications

A working version of the program (***.exe*** file only) is downloadable. See the link to it above. It is called ***leasing_ui.exe*** (stands for leasing user interface). [Link is available on the main assignment page.] This version shows the user interface you should use; the program itself is fully functional.

You are to develop a form laid out just like the one in the working version, complete with the same controls, labels, titles and colors. Please be careful of your sizing of the form and controls and the spacing; sizing and spacing should approximate that in the working version.

The form should have a fixed single border style (meaning the borders are NOT resizable). Each of the items in the chart (including building) adds a discrete positive or negative amount onto the base rent.

Base rent can be any number. No validation checks are needed for this program. You will assume the user is entering a positive integer value.

Building is a combo box (DropDownList style) that has the two buildings specified above. The buildings should appear in the dropdown list in the same order as they do above. However, you should assume **SunView**; it should be the default building which appears in the box upon running the program. Remember, this is a dropdown list, so the user should not be able to add any value not on the list, nor can the user "character-edit" any value which appears in the list. Renting in Crashing Surf adds $210 to the rent.

Lanais are available in most, but not all, units. A $65 per month premium applies. This item should already be "checked" upon entering the form. This means that if the base rent is $1000, the final rent displayed (without changing ANY of the default values) is $1065, since Lanai is checked.

A few units have a rooftop desk. A $120 per month premium applies. This item should be "unchecked" upon entering the form.

Some of the units are non-smoking, and these suffer less wear and tear than other units. There is a $25 per month discount on these. This item should be "unchecked" upon entering the form.

The final rent box (for display of the final rent) should have the slight 3D sunken look, but because it is only for display, it should NOT be a text box but a label. ***Important:*** When you create the label, one of the properties (**AutoSize**) will be set to **True**. You will need to change this to **False** so that you can size the label appropriately.

Also, think about property in this control you'd need to change to get the border to have the Fixed3D look. Also, the text should be aligned to be centered (vertically and horizontally) within the label. (This last statement gives a hint which property controls the alignment of the text. Figure it out.)

The label above the final rent box that identifies it should be bold and slightly larger as shown. The amount that appears inside the box should be bold and large (14 point or so) as shown. Do not display any $ or decimal places in the box. The box should be empty upon program start.

The base rent text box should be blank upon starting the program. The default building which should be displayed is **SunView**.

Note that the form itself should have the title "Haleakela Lofts Leasing Assistant" appearing in the title bar. Also, you should definitely include the labels you see, including the name of the complex and its slogan. Everything should be formatted and aligned as you see in the sample.

The Exit button should end the program (otherwise known as "closing the form"). The Calculate Rent button always needs to be pressed to determine the latest rent.

Please do not vary from these specifications or your score will suffer!

Hints for Development

Students should direct their Windows systems to always display file extensions for this course. This is in the Windows Explorer under Tools, Folder Options, View. Uncheck **Hide File Extensions for Known File Types**.

Please save the form on your hard disk with the filename of **frm** plus your **first initial and last name** plus **_hl** and the extension of **.vb**. So for me, it would be ***frmKJefferies_hl.vb***. For Jane Smith, it would be ***frmJSmith_hl.vb***. Remember also give the form an internal name; you must use **frmLeasing.** Remember also that you need to go to the Project/Solution Properties (as shown in class) after saving and make sure that **frmLeasing** is the "startup" object.

Remember to set initial properties using the properties window. You may also choose to explicitly set these options to their defaults by supplying some code in the Form_Load event procedure, If your form is called **frmLeasing**, then the following would be the event procedure that executes upon the loading of the form: **frmLeasing_Load( )**. You will need to do this to set your default area combo box. Others may be done directly in the Properties window.

Development in Visual Basic is very different from C or other procedural languages. You can test your creation at any time by pressing F5 and running it. Do a little at a time. You will end up with several subroutines, but that's the nature of Visual Basic. All of your subroutines will be Private Sub. You will NOT be using multiple forms, Visual Basic modules, or Public subroutines. Please keep it simple. You really shouldn't need anything that wasn't discussed in the lecture.

Try to come up with a coherent and <u>simple</u> algorithm for determining the rent. It is much easier to let your Calculate Rent button do the major work of the program than to trigger actions based on selecting one of the rental options. The important thing? It had better work and give the right answer!

Use the appropriate controls for the appropriate action. Base Rent is an input, so you will use a textbox for it. The final rent is just an output, so you will use a label for it. I'll let you examine the various properties to see how to get that "3-d recessed" look and color for the label (there’s not too many choices, so you should be able to figure it out!)

Please use good commenting for full style points. Name your controls other than the default (in other words, don't use Text1, Combo1). Give meaningful names and use the naming conventions we discussed in class. Your algorithm should be simple and straightforward for full style points. If you end up doing the program in a way which creates a whole bunch of selection statements or a big giant linear-nested if, then you are NOT doing the program the simple and direct way! Consider this: If the specs were to change to add two more checkboxes on the form, would you program add just a couple selection statements? Or would you end up quadrupling the amount of code you have? If your answer is the latter, then you are approaching the program the wrong way. **Hint:** Don't think like what you think a "computer" thinks like. Think like a human. If you were looking at a paper form, how would YOU calculate the rent? If you approach the program the way most people would calculate, you will have an easy time of developing this.

Because we have not yet covered variables or defined constants, you do NOT need to use variables or defined constants to do this assignment. You can essentially use the text property of your textbox or label as a "variable". **In fact, I'd like to specifically ask you to NOT use variables in this program!** Let's keep it as simple as possible.

Don't worry about a formal design for this program; you won't be handing one in. HOWEVER, you should still do some design work. After doing the user interface, look at each control and identify to what events you'll be doing something in response. (Generally, for this class, it's Click). Then use some form of pseudocode or N-S chart for EACH of these event procedures.

Make sure you "clean up" your code before submission. It should look good, and not have a bunch of "empty" event procedures showing (where you erroneously double-clicked on controls and created these procedure shells). Leaving "empty" event procedures in your code will result in a style point reduction.

What and How to Submit

Your final submission to me should be **ONLY** your two Visual Basic form files. Those are the files (one of which was mentioned above) that contains your name. So as an example, if your name is Karen Smith, you would submit the following **two** files:

- ***frmKSmith_hl.vb***
- ***frmKSmith_hl.Designer.vb***

I will load these files into my own new project, build and run it, and test it for accuracy and proper operation. You will receive up to 5 points for this assignment.

Critical—read carefully: Please submit only these two ***.vb*** files via the assignment submission area on the Canvas web site for our course. Please make sure you include both these files (and these files only!); I need both of them to be able to build an executable program. Without both, your submission is incomplete and I can't possibly grade it (which means you will receive a zero). Also if you submit a different file (the wrong one, such as the .exe or

.resx or .vbproj) you will receive a zero for the assignment. None of these files have any form information or program code, so do not submit them.

Both files need to be submitted during the same submission session, and this should be done PRIOR to the due date/time.

Visual Basic doesn’t take kindly to changing filenames outside of the development environment, so if you need to rename your files, please do so using the Solution Explorer in the VB development environment. It is similar to changing a filename in Windows Explorer.

Please check the announcements for the due dates and announcement of any spec changes and clarifications.

Words to the Wise (An important re-statement and re-emphasis of what I mentioned above)

Start working on this program immediately with your current knowledge. It should honestly take less than 30 minutes to design the form, and less than that for the code. Really! The algorithm for this program is very very simple, and the code for this program should only be about 35 lines, and that's including some spacing and comments.. If you are writing dozens of lines of code, you are doing this program the wrong way (and will suffer a bit in your style grading). If you find that you are using a gigantic linear nested-if statement, you are also doing this the wrong way. Think about how a human might “process” the form manually, and that will clue you in to the easiest way of doing the program.

This is your first Visual Basic program. But you've been through some programming now and see what's needed to map the real world into some code. You saw in class the Visual Basic syntax is fairly easy, and this program uses only the most basic of statements. The key is here to pay attention to the details. Get the numbers right; that's of utmost importance. Secondly, get the user interface right. Test your program! For those things that are a bit sticky, if you need a nudge, send me an email via Ask Your Instructor in Canvas and I'll try to help you out.