Assignment 6 - Visual Basic, comprehensive

for Visual Basic - revised 3/2020

General Information

You are to develop a lease pricing app for the Desert Vista Country Club, a beautifully landscaped rental housing complex with affiliated golf course located in the Coachella Valley. The non-smoking complex sports views of both the San Jacinto and Santa Rosa mountains. The complex is divided into three areas: **Palm Island, Moderne and Tahquitz,** each of which has its own unique character and style of units. The app allows the user to enter a base rent (which would be the minimum rent charged for a standard version of the least expensive unit type) and then to select a variety of attributes about a rental unit, include:

- type of unit
- number of bedrooms
- number of bathrooms
- view
- location (area in complex)
- pet
- other amenities, including pool proximity, golf cart parking space, lanai

The combination of attributes determines the ultimate monthly rent to be paid. Moreover, the program figures out the deposit amount too, based on type of pet, credit rating of the renter, and other factors.

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

You are to develop this program using the version of Visual Basic that you have been using in this course. The program should be written at the level of the course, using VB features and techniques from the class. It should not use VB syntax that we haven't covered.

Functional Specifications

As you start this assignment, you will be given images of the form. You will not be given a functional example of the program for another week, at which time an .exe file will be downloadable. This sample version shows the basic user interface you should use.

Although the sample program itself will be reasonably accurate, it is not 100% accurate. It has not been fully tested and the rent and deposit amounts shown may be incorrect in certain instances other than with the grading criteria data. After failing validation, the focus should be located back to the text box for Base Rent and the box should be cleared. The access keys are, though, fully represented in this interface.

Please use the functional specifications below to determine actual function along with the JPG images found outside this document on the Canvas page, unless otherwise noted.

Static pictures of the form are available on the Canvas page for the assignment. You will need to scroll to see them. Please view it if you are having trouble running the sample program.

Form specs:

- Form should be laid out just like the one in the sample version, complete with the same controls, and should NOT be re-sizable when running. 815x440 pixels is the size.
- Control locations and control/form sizes should be approximately the same.
- The form itself should have the title **Desert Vistas CC Leasing App** appearing in the title bar. Also, you should definitely include the other labels you see on the form itself, including the name of the complex and its slogan.
- Access keys should be specified for ALL options on the form, including those for the **Base Rent**, **Area**, **Calculate** button, **Exit** button and menu (menu title and menu items).
- Tab order should follow the functional example given to you in the second week. Tab order must be established properly for many of the access keys to work.

- The program’s executable build uses **XP Visual Styles** (see the middle of the project's properties page), so the program should appear close to the same on what you build. There's no reason to panic or contort the look of any of the controls; the program basically uses the “natural” look of each control

- Most of the items in the form adds onto either base rent, deposit or both. Selection of some options excludes the availability of other options, and these will be noted below. The view specifies a multiplier to a standard rent charge (called Rent Before Amenities, or RBA), based on which view is selected. Final rent is defined as the RBA plus all amenities.

- You will note several group boxes. Please be aware of the instructions below for creating option groups, as each group is exclusive of the others.

Base Rent Specs:

- The base rent can be any number from 1000 up.

- If it is less than 1000, a Critical-style message box should be displayed indicating the error.

- The error text should read: **Base rent must be 1000 or higher**, the box title should say **Input Error** and the button should be **OK**.

- If the rent is non-numeric, a Critical-style message box should be displayed indicating the error.

- The error text should read: **Base rent must be a number**, the box title should say **Input Error** and the button should be **OK**.

- This minimum could change every few months. Design your program so that it uses a defined constant for the minimum possible base rent, as well as for the initial base rent.

Area Specs:

- Area is a group of radio buttons that enumerate the following specified below.

- Tahquitz is the oldest and largest area of the complex, containing only traditional 1- and 2- bedroom apartments.

- Palm Island units are more luxurious and integrated with the golf course, although not all Palm Island units have views. Both 1- and 2-bedroom apartments are found on Palm Island, along with Casitas which are standalone open floor plan 1-bedroom units).
- Moderne is the newest area of the development. Its architectural design is similar to the Mid-Century Modern style found in Palm Springs. Only lofts are located in Moderne.

Unit Bedroom Specs:

- 1-bedroom apartments have the base rent as their rent, before accounting for other factors.
- 2-bedroom apartments are $150 higher than the base, before accounting for other factors.
- Casitas are always $96 more than the base rent, before accounting for other factors.
- Lofts are expensive, costing twice the base rent, before accounting for other factors.

Unit Bathroom Specs:

- Units have either 1 Full Bath, 1 ½ Baths or 2 Full Baths, depending on the style of the unit.
- Casitas and Lofts never have more than 1 Full Bath, so selection of either should select 1 Full Bath and disable the other bathroom options.
- Other 1-bedroom units never have 2 Full Baths, so selection of it should disable two full baths. By default, they have 1 Full Bath (unless the other option is already selected when changing from a bigger unit).
- 2-bedroom units never have only one bathroom; they always have 1 ½ Baths or 2 Full Baths, one and a half baths or two baths, so disable 1 Full bath if 2- bedroom is selected. By default, they have 1 ½ Baths.
- <u>Remember always to re-enable the options for other size units if the user selects something different.</u>

- For 1-bedroom units, having 1 ½ Baths adds a $60 monthly premium over the standard 1 Full Bath.
- For 2-bedroom units, having 2 Full Baths adds a $37 monthly premium over the standard 1 1/2 Baths.

View Specs:

- View pricing is on a multiplier, which raises the cost of the unit's RBA (Rent Before Amenities--such as Golf Cart Space, Pool Proximity or Pets). Units are charged on the best view that they have.
- View is a combo box of DropdownList style, with the order specified as shown.
- Parking views are standard and the default when entering the form.
- Greenbelt views command 6% more than the RBA.
- San Jacinto Mountains command 15% more than the RBA.
- The best views overlook the **Golf Course**, **Lakes** and/or the **Santa Rosa Mountains** to the south. A 33% premium over the RBA applies to units with any of these views. If a unit has two or more of these views, these leasing office tags it only with one of the "best" of its views. For example, a unit with a Greenbelt view and a Lakes view will be tagged as having only a Lakes view. A unit with both a Golf Course view and a Lakes view will be tagged as having either a Golf Course view or a Lakes view for pricing purposes. (In other words, in the app, only a single view is listed for a unit.)
- Moderne is on the north side of the complex and none of the units have any of the Golf Course or Lakes views, so these two views cannot be selected when Moderne is selected as an area. Upon the user selecting Moderne, the program should change the view to San Jacinto Mountains. And in the future, if Moderne is the area and the user tries to select either Golf Course or Lakes for a view, a MessageBox should be displayed with the text "View not available from Moderne" and window caption "User Input Error", and then San Jacinto Mountains view should be automatically selected.

Amenities Specs:

- These items apply <u>after</u> RBA, so they are not part of the multiplier.

- Almost all units have a Lanai, so not having one is rare and would constitute a $42 per month discount. This item should already be "checked" upon entering the form.

- One pet is allowed in a unit (dog or cat), but there is a monthly charge of $25, In addition, $200 is added to the Deposit.

- Golf Cart Space is $70 per month, and is subject to availability. Like other amenities, Golf Cart Space is NOT part of the RBA (therefore not subject to view multipliers) and is the same price in all areas of the complex. It should, though, be included in the on-screen rent display and it does add $50 onto the deposit.

- Pool Proximity applies to units adjacent to pools, and adds a flat $10 to the rent after RBA.

Deposit Specs:

- The deposit is generally equal to one month's final rent (after applying all options and multipliers to the RBA, plus amenity charges).

- Remember, a pet adds $200 to the Deposit.

- For renters with bad credit, the entire deposit (including premiums, and after rounding to the nearest dollar) is doubled.

Display of Rent and Deposit Specs:

- Both rent and deposit boxes have 3D "sunken" look (and the user should <u>not</u> be able to tab or click into them), so be sure to pick the right control and properties for this.

- Try your best to match the colors; their names are Aqua and Yellow.

- The labels for them should be bold and 12 pt as shown, centered over the display box below them.

- The amounts inside both should be bold and large as shown (16 pt), and placed in the center and middle of each control.

- Amounts should show a $ in front of the number and be formatted with a comma for numbers over 999 (American currency style). Remember the Module 9 lecture that discusses Parameters? Or look it up!

- The final rent amount and deposit amount should not have any decimal points or decimal values and should be <u>rounded</u> to the nearest integer value. (How do you round in Visual Basic? You get to look up this function too!)

- Both boxes should be completely empty upon program start.
- If either of the error conditions described above occur (base rent less than minimum or base rent being non-numeric), both the rent and deposit box should display the word **Error** (capital E lower case for rest of the word) upon the user clicking the Calculate Rent button and the appropriate Input Error message box should then pop up on-screen. Once the user clears the message box, **Error** should still display in the rent and deposit boxes.

Defaults Upon Entering the Form Specs:

- The base rent box should contain the minimum possible base rent (see above) upon starting the program.
- The default area which should be displayed is **Tahquitz**. Order of the area combo box is **Tahquitz**, then **Palm Island**, then **Moderne**.
- The default bedroom size is **1 Bedroom**.
- The default bath is **1 Full Bath**.
- Any option not available to **Tahquitz** or **1 Bedroom** units should be disabled upon start of the program (to be re-enabled later if the appropriate option is selected).
- The default view is **Parking**.
- Lanai should be checked.

Menu Specs:

- Your program should incorporate a simple Windows menu. The menu title should be **Specials**. Use "S" as the access key for this. (Remember, users typically will NOT see the underscore during execution until they press ALT.)
- The items on the menu should be: **Cozy Lovenest, Executive Cool Pad** and **Exit**. Use the designated access keys.
- Use one of those thin menu separator lines separating the two featured scenarios from the Exit menu item.

- Selecting **Cozy Lovenest** sets the following on the form:
    - Palm Island
    - Casita
    - Lakes view
    - no lanai
    - no pets
    - any other options (the check boxes) should be unchecked
- Selecting **Executive Cool Pad** sets the following on the form:
    - Moderne
    - San Jacinto Mountains view
    - lanai
    - pet
    - Golf Cart Space
    - any other options (the check boxes) should be unchecked
- Pay close attention to all access keys on the form. Define those access keys specified in the sample interface.

Other Specs:

- When changing from one of the bedroom specifications to another, if certain bath sizes (or other such attribute) are not available to the new specification, then those that are not must be disabled. Moreover, the default bath size (or other such attribute) should be automatically selected at that time.
- "Disabling" a radio button does NOT mean making it invisible. It should only be grayed out (typically, by setting the Enable property to false).
- The **Exit** button should end the program.
- The **Calculate Rent** button always needs to be pressed to determine the latest rent and deposit.

- Both buttons use access keys; see the interface for more information.
- It is possible for the final rent to be BELOW the base rent. For instance, selection of the least expensive unit that also has no lanai would do this, due to the fact that this item is a discounts from the rent.

Clarifications

Again, it is possible to have the calculated rent be BELOW the base rent under certain circumstances. For instance, not having a lanai in a 1-bedroom makes calculated rent less than base rent.

This is okay. Base rent is meant to be an index. As long as the user of the program knows, then it's okay. If the program had help, I'm sure we'd put that in there!

All sizes of controls and form locations should be approximated. Please stick to MS-Sans Serif font for everything. **The form itself should <u>not</u> be resizable**.

Hints for Development

Remember even though you can set some initial properties using the properties window, you may also explicitly set these options to their defaults by supplying some code in the Load event procedure for the form. This is a more professional way of doing it.

To create the numerous radio buttons on the form, it is necessary to draw the group box first before creation of the radio buttons that go inside it. Test these in the interface, making sure you've placed your buttons correctly and that they are properly "bound" to the appropriate group box. This doesn't take any special command, just proper graphic placement.

Development in VB 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 lots of event procedures, but that's the nature of VB.

I want lots of comments and defined constants. Plus good modularity with no super long subroutines.

Try to come up with a coherent algorithm for determining the rent. Some of you may try to do the calculations all when the button is clicked. Others of you will change the values of certain "offset amount" variables as you change the options. It's up to you (although the first way is probably easiest for most of you). The important thing? It better work and give the right answers!

This is a program that CRIES for code subroutines (other than the many event subroutines), defined constants, and good commenting. Make sure you do so!

You should also try to distinguish properly between text boxes, labels and actual variables. Use variables for your calculating, text boxes for input, and labels for display. Remember, you can make a label look like a textbox (except it doesn't allow users to get the value); please do this when necessary. Name your objects other than the default (in other words, don't use Label1, Combo1). Give meaningful names and use the naming conventions we discussed in class.

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. I'd use some form of pseudocode or Nassi-Schneiderman chart for each event procedure just to indicate what the program is to do when.

Assignment 6 Naming Conventions

Please use the following conventions for naming your Assignment 6 form. The file name is based on a fictional student named Jenny Chen, so you should use your own first initial and last name.

Internal form name:	**frmDesertVistas (only the D & V are capitalized)**
File name example:	**frmJChen_dv.vb**

The designer file will be created automatically, and will be called (as an example): **frmJChen_dv.Designer.vb**

These naming conventions must be followed for this assignment, or your points will be affected.

What and How to Submit

Your final submission to me should be your form file (.vb file) AND your designer file (Designer.vb file). We will run the application and test it for accuracy and proper operation. You will receive up to 5 points for this assignment.

Please submit only the specified file(s) via the assignment submission area on the Blackboard web site for this course. Do not submit any of the other files generated. Do not email the files to me. If you need to resubmit before the due date / time, please only submit both files together. Try your best, however, to submit only one final set of files. The last pair submitted prior to the due date is the one that will count for the grade. Please check the announcements for the due dates and announcement of spec changes and clarifications.

PAY CLOSE ATTENTION TO THE FOLLOWING: DO NOT SUBMIT THE **.vbproj** FILE OR **.sln** file OR ANY OF THE OTHER FILES GENERATED. YOU MUST SUBMIT THE **.vb** FILE (representing the form) AND THE **.Designer.vb** file BY THE DUE DATE OR YOU WILL RECEIVE A SCORE OF ZERO FOR THIS ASSIGNMENT. I cannot execute a **.vbproj** file or **.sln** file or any of the others because they don't have code in them. And while I can run an **.exe** file, I can't grade it because there is no code to grade.

Words to the Wise

This is your first "big" Visual Basic program, and yes it is quite "rich" and full of details. But you've been through some C and VB programming now and see what's needed to map the real world into some code. You saw in class the VB 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, test, test, test, test your program! For those things that are a bit sticky, if you need a nudge, send me an email and I'll try to help you out.