CS2453: Visual Basic
VB Assignment 3: Long Distance Calls

Objectives

- Effectively use relational and logical operators.
- Create EFFICIENT selections structures.
- Validate input including numeric, range, and data types.
- Use string methods and functions.
- Create message boxes.
- Create and code radio buttons and checkboxes.

Situation

See p. 282, "10. Long Distance Calls" for the basics of the assignment. In addition to the data required by the book two additional input fields will be included to obtain an email address and then to retype the email to confirm a correct entry. The email address will be validated to make sure it includes a @ (at sign) and a . (period). Then the two emails will be compared to make sure they match. The comparison will be case insensitive. When the total cost is displayed, the email address to which the bill will be sent will also be shown. TryParse must be used to validate numeric input. Only whole numbers (integers) are to be accepted.

User Input		Program Output
Rate Category	**Minutes**	**Monthly Charge**
Daytime	20	$1.40
Evening	20	$2.40
Off-Peak	20	$1.00

The program will also be enhanced so that when there is an error in an input field, the cursor will be returned to that location and the incorrect input will be highlighted.

Specifications

1. Recurring Specifications that are required for all programs.
    1. The form must be renamed and the text changed to Long Distance Calls by YourFirstName YourLastName. (If Pat Programmer was creating this

program, it would be **Long Distance Calls by Pat Programmer**)

2. Code must be grouped and commented in compliance with this course's programming standards.
3. ALL files, forms, and controls MUST be renamed.
4. Option Strict and Option Explicit must be ON
5. Controls must be locked.
6. An AcceptButton and a CancelButton must be assigned appropriately.
7. ALL controls on the form must be in logical TabOrder.
8. All buttons and labels (before TextBoxes) must have AccessKeys.
9. Form's StartPosition property must be CenterScreen.
10. Values from the input fields MUST be assigned to variables and the variables used in calculations.
11. The text property of Labels must be changed so that Label1 (or similar name) does not appear at runtime.
12. No class level variables unless specifically allowed.
13. Data types for variables and constants must be the most efficient.

2. Use a minimum of at least one GroupBox to arrange controls as needed.
3. Create three buttons: Calculate, Clear, and Exit. Write appropriate code for each button. For the Clear and Exit buttons, include MessageBox functionality which gives the user a choice of continuing or canceling. The MessageBoxes must include icons.
4. Create three input areas: minutes used, email, and confirm email. Validate these as follows
    - Hours -
        - Numeric validation - use TryParse
        - Range Validation – 0-20000 minutes.
        - Provide an appropriate error messages depending on the error
        - Set focus to input area with the error.
    - Email
        - Validate email by checking for a @ and a . in the email address (It is realized that this is not a very comprehensive validation, but it will provide experience processing strings)

- Validate that email and confirm email are equal. Allow for variations in case. (Example: PProgrammer@gmail.com and pprogrammer@Gmail.Com would be considered equal)
- Provide an appropriate error message depending on the error
- Set focus to the input area with the error.

5. On p. 282, all rate values must be declared using constants. In addition the maximum number of minutes (20000) must also be a constant.
6. A label will display the results. You are to display in the label, the category selected, the minutes used and the total bill on one line. A sentence stating the bill will be sent to the provided email address MUST be on a new line. The email address has to be displayed. The cost must be formatted as currency.
7. Coding must be EFFICIENT. It is suggested that you flowchart the validations and calculations to determine all of the possibilities before actually writing the code.
8. Various erroneous inputs will be tested during the grading process. If any of them result in the program halting of crashing, the **maximum** credit will be 50%. If submitted by the initial deadline, it may be resubmitted until the final deadline. Work that is late or resubmitted is subject to a 10% penalty, but that might be better than 50% or a 0.

Assignment Submission

Zip the entire Internet Service Provider folder and submit it using Moodle

Program Grading Criteria

Grading Chart

Item	Possible Points	Points Off	Comments
Recurring Specifications (13 items listed above under	10		

Specifications)			
At least one GroupBox, more if appropriate.	1		
Use of RadioButtons	4		
Use of MessageBoxes (including buttons options with actions based on user's choice)	3		
Correct naming and use of Constants.	3		
Validation of Minutes used	5		
Validation of Email and Confirm Email	5		
Coding for Clear button including a MessageBox	3		
Coding for Exit button including a MessageBox	3		
Coding for Calculation button	10		
Total Cost message including Rate Category type, minutes used, charge formatted as currency, newline, and email message.	5		
Interface design, spelling and grammar	3		
Efficiency of code	5		
Total	60		
Late/Resubmit (10%)	-6		
Final Total			