Use the following test data to determine if the application is calculating properly:

|  |  |  |
| --- | --- | --- |
| **Pallet** | **Pallet and Widgets** | **Number of Widgets** |
| 100 | 5,620 | 600 |
| 75 | 1,915 | 200 |
| 200 | 9,400 | 1,000 |

**Task 1: Design the Application**

1. The program should also include a Clear button that clears the controls used to display input and output and an Exit button that closes the application. The application should display a Welcome message in a dialog box when the application starts. The welcome message should be: “Hello. The time is *current time*.”
2. In Microsoft Word, create a table that describes the characteristics of the application.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Features** | **Inputs** | **Calculations** | **Conversions** | **Outputs** |
| Welcome Message  Input boxes  Result Message  Clear, Calculate & Close Buttons | Pallet Weight  Total Weight of  Pallet & Widgets | (Total Weight  Minus  Pallet Weight)  divided by  constant 9.2  equals  Number of Widgets on Pallet | Textbox strings to Integers  Decimal to Integer  Integer to String | Calculated Number of Widgets |

1. Use Microsoft PowerPoint to sketch the user interface for the application.

Tip: Use GroupBox to form a frame around the all controls except the Close button.

1. Copy the sketch and paste it into the Word document.

“Hello. The time is *current time*.”

Clear

Exit

Number of Pallets

Number of Widgets

100

600

Pallets & Widgets

5,620

Widget Weight

9.2 lbs

1. Create a table that lists the type of control and the name. Make sure to use standard naming conventions and use names that adhere to Visual Basic naming rules.
2. Add a column to the table and list the default Text property for each control.

|  |  |  |
| --- | --- | --- |
| **Control Type** | **Name** | **Default Text Property** |
| Label | lblWelcomeMessage | Hello. The Time is 12:00 PM. |
| Label | lblPalletWeight | Pallet Weight: |
| Label | lblTotalWeight | Pallets & Widgets |
| Label | lblNumberOfWidgets | Number of Widgets = |
| Label | lblWidgetTotal |  |
| Textbox | txtPalletWeight |  |
| Textbox | txtTotalWeight |  |
| Button | btnClear | Clear |
| Button | btnCalculate | Calculate |
| Button | btnClose | Close |

1. Create a table that lists the event procedures the program requires.
2. Specify the event procedure name and a description of what it does.
3. List each variable the event procedure will use. Define the name, data type, and initial value.
4. List any constants you will use in the event procedure. Define the name, data type, and value.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Event Procedure | 1 | 2 | 3 | 4 |
| Name | Form1\_Load | btnClear\_Click | btnClose\_Click | btnCalculate\_Click |
| Description | Sets welcome message when form loads then places cursor in Pallet Weight textbox. | Clears all fields when the Clear button is clicked on. Resets welcome message. Places cursor in Pallet Weight textbox. | Closes application when Close button is clicked on. | Calculates number of widgets on pallet and displays answer in label. If either text entry field is blank, sets focus to empty box and selects added message “Weight?”.  Resets welcome message. Places cursor in Pallet Weight textbox. |
| Variable 1  Name  Data Type  Initial Value | None | None | None | Procedure-wide  palletWeight  Integer  0 |
| Variable 2  Name  Data Type  Initial Value |  |  |  | Procedure-wide  totalWeight  Integer  0 |
| Variable 3  Name  Data Type  Initial Value |  |  |  | Procedure-wide  widgetWeight  Decimal  0 |
| Variable 4  Name  Data Type  Initial Value |  |  |  | Local  calcWeight  Integer  CInt(widgetWeight) |
| Constants  Name  Data Type  Value |  |  |  |  |

1. Save your Word document.

**Task 2: Create and Test the Application**

1. Start **Visual Studio**.
2. Create a new Windows Application project named **Module2\_Lab1\_Widget**.
3. Set the title of the form to “**Widget Calculator**.”
4. Add the controls you identified in Task 1 to the form. Set their positions to match your sketch. Lock the controls when you are satisfied with your design.
5. Set the Name and Text properties for each control to match your design.

***Question 1:*** *How would you set the Text property of a label to display “Pallet & Widgets”?*

***Answer 1:* Pallets && Widgets**

1. Implement keyboard access.
2. Make the Clear button respond to the Escape key.
3. Make the button that performs the calculation, respond to the Enter key.
4. Add code to set Option **Explicit On**.
5. Write an event procedure to implement for the Load event procedure of the Form.

***Question 2:*** *What object and method will you use to display the Welcome message?*

***Answer 2:***

Private Sub Form1\_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles MyBase.Load

lblWelcome.Text = "Hello. The time is " & DateTime.Now.ToShortTimeString() & "."

End Sub

1. Write an event procedure to implement the Clear button.

***Question 3:*** *Which technique did you use to clear the input and output controls?*

***Answer 3:***

Private Sub txtUser\_KeyDown(sender As Object, ByVal e As KeyEventArgs) Handles MyBase.KeyDown

If e.KeyCode = Keys.Escape Then txtPallets.Clear()

If e.KeyCode = Keys.Escape Then txtWidgets.Clear()

If e.KeyCode = Keys.Escape Then txtTotal.Clear()

End Sub

Private Sub btnClear\_Click(sender As Object, e As EventArgs) Handles btnClear.Click

txtPallets.Clear()

txtWidgets.Clear()

txtTotal.Clear()

End Sub

1. Write an event procedure to implement the Close button.
2. Write an event procedure to calculate and display the number of pallets, based on the weights entered.
3. Display the result as a whole number.
4. Use exception handling to display the error message of the exception if the user enters a value that causes an exception. Tip: Use the Message property of the exception.
5. Use comments to document your code.

***Question 4:*** *What data type conversion function or functions did you use?*

calcWeight = CInt(widgetWeight)

calcWeight.ToString()

1. Test with the values given with the scenario.
2. Test by entering non-numeric values for the weight of the pallet or the weight of the pallet and widgets to verify that your exception handling code works.
3. Save the project when you are finished.
4. Turn in your written answers to the lab questions.

**Part 2:** Perform the following tasks:

**Task 1: Design the Application**

1. Read Programming Challenge #6 given on page 350 of the textbook. The program you create will be similar, except that you will allow users to enter the number of stores in a TextBox control. You will use input validation to ensure that the number of stores is a number greater than 0. If a user clicks Cancel in the dialog box, you should prompt the user whether to continue entering data or calculate the bar chart. If the user enters invalid data, you should display an error message to the user. Do not show the button to display the chart unless the TextBox control for the number of stores contains a valid value. If the user enters invalid sales data, an error message should display and the user should be allowed to reenter data for that store. If sales are less than 100, you should not display an asterisk.
2. Make a list of the controls you will add to the form. Include the Name property.

List of controls to add to the form:

|  |  |
| --- | --- |
| Controls | Name |
| Label | lblStores |
| Label | lblSales1 |
| Label | lblSales2 |
| Label | lblSales3 |
| Label | lblSales4 |
| Label | lblSales5 |
| Textbox | txtStores |
| Textbox | txtSales1 |
| Textbox | txtSales2 |
| Textbox | txtSales3 |
| Textbox | txtSales4 |
| Textbox | txtSales5 |
| Button | btnCancel |
| Button | btnDisplayChart |
| Button | btnExit |

1. Make a list of the event procedures you will code.
2. Use Visio to create a flowchart for each event procedure.

***Question 5:*** *How many loops did you use?* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Save the Visio document.

**Task 2: Implement the Bar Code Application**

1. Launch Visual Studio 2012.
2. Create a new Visual Basic Windows Forms Application project name Module2\_Lab1\_BarCode.
3. Add the necessary controls to the form.
4. Set the Name property of each control.
5. Set any other properties, as required by the design.
6. Write the code to implement data validation. Make sure to document your code, using comments.

***Question 6:*** *How can you make it easier for a user to correct a data entry error?*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Test your code with valid and invalid values.
2. Write the code to implement the Exit button.
3. Test your code.

***Question 7:*** *Can you exit when the TextBox control contains invalid data? If not, why not?*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Stop Debugging.
2. Add the following code to the btnExit\_Click event procedure. Place the code immediately before you call the Close method:

txtNumberOfStores.Text = 1

1. If you have not already done so, set the CausesValidation property of the Exit button to False.
2. Test your code again.
3. Add the code to the Click event of the button that displays the bar graph. Use the hints in the table:

|  |  |
| --- | --- |
| **Code part** | **Hints** |
| Outer Loop | a. Use a Do While....Loop.  b. Use Exit Do to exit, if the user does not want to continue.  c. Make sure to initialize and increment the counter.  d. Test whether the input is numeric, then use AndAlso to test the range. |
| Inner Loop | a. Use a Do Until...Loop.  b. Use integer division to determine the number of times the loop should execute.  c. Make sure to initialize and increment the counter. |

1. Add at least one comment to your code.
2. Test your code with the sales values shown in the table. Set breakpoints to step though your code and debug, if necessary.

|  |  |
| --- | --- |
| **Value** | **Expected Result** |
| 0 | Error message |
| “No sales” | Error message |
| 50 | No asterisks |
| 100 | \* |
| 102 | \*\* |
| 500 | \*\*\*\*\* |

1. Turn in your written answers to the lab questions.

**Submission Requirements:**

Submit the Word document titled “SD3320\_Module2\_Lab1\_Part1.docx” and “SD3320\_Module2\_Lab1\_Part2.docx” to your instructor for grading. Make sure that you also include a screenshot that shows that the application developed is working correctly. In addition, make sure to submit the project files titled “Module2\_Lab1\_Widget” and “Module2\_Lab1\_BarCode.” The Word document should have the following specifications:

* Font: Arial; font size: 12; double-spaced
* Length: 1 page

**Evaluation Criteria:**

Your performance for Part 1 will be evaluated against the following points:

* Does your document list the characteristics of the application?
* Does the program display the correct output?
* Does the program display the correct error message when the user attempts a calculation with invalid input?
* Does the program clear the input and output fields?
* Does the program display the correct Welcome message when it starts?
* Does the program quit when the user clicks Close?

Your performance for Part 2 will be evaluated against the following points:

* Did the Display Graph button display only after typing a valid number in the Number of Stores field and causing the field to lose focus?
* Were you able to reenter data for Store 1 after entering invalid sales data?

Did the test data generate correct results?