Algorithm

1. Add a String to ComboBox from TextBox

Algorithm:

Step 1: Create a Windows Forms application.

Step 2 :Add a ComboBox (ComboBox1), TextBox (TextBox1), and Button (Button1) to the form.

Step 3 :In the Button click event:

Retrieve the text from TextBox1.

Step 4 :Add the retrieved text to ComboBox1.

Clear TextBox1.

2. Display Hierarchical Items in TreeView

Algorithm:

Step 1 :Create a Windows Forms application.

Step 2:Add a TreeView control (TreeView1) to the form.

Step 3:In the Form Load event:

Create a root node (e.g., "Root").

Create child nodes (e.g., "Child 1", "Child 2").

Add child nodes to the root node.

Add the root node to TreeView1.

3. Handle User-Defined Exceptions

Algorithm:

Step 1:Create a custom exception class that inherits from Exception.

Step 2:Create a Windows Forms application.

Step 3:In the Button click event:

Use a Try-Catch block.

Step 4:Throw an instance of the custom exception with a message.

Step 5:Catch the exception and display its message in a MessageBox.

4. Employee Details Using Constructors and Member Functions

Algorithm:

Step 1:Define an Employee class with properties (e.g., Name, Age).

Step 2:Implement a constructor to initialize these properties.

Step 3:Implement a method to return employee details as a string.

Step 4:Create a Windows Forms application.

Step 5:In the Button click event:

Instantiate an Employee object with sample data.

Display employee details using the method in a MessageBox.

5. Demonstrate Various Events

Algorithm:

Step 1 :Create a Windows Forms application.

Step 2:Add controls (e.g., Button, Label).

Step 3:Implement event handlers for:

Form Load: Show a message indicating that the form has loaded.

Step 4:Button Click: Show a message indicating that the button was clicked.

Step 5:Mouse Down on Form: Show a message when the mouse is clicked on the form.

Step 6:Key Down on Form: Show which key was pressed.

6. File Menu with Menu Items

Algorithm:

Step 1:Create a Windows Forms application.

Step 2:Add MenuStrip control to the form.

Step 3:Define File and Edit menus with respective items (e.g., New, Open, Save).

Step 4:Implement click event handlers for each menu item to perform actions (e.g., show messages or open dialogs).

7. Student Information Database Operations

Algorithm:

Step 1:Set up database connectivity (e.g., using ADO.NET).

Create methods for:

Step 2:Adding student information to the database (INSERT).

Step 3:Deleting student information from the database (DELETE).

Step 4:Updating student information in the database (UPDATE).

Step 5:Create a Windows Forms application with controls for user input and buttons for each operation.

8. Web Form to Show Current Date and Time

Algorithm:

Step 1:Create an ASP.NET Web Forms application.

Step 2:Design a web form with:

A Label control to display date and time.

A Button control to trigger the display action.

Step 3:In the Button click event:
Retrieve current date and time using DateTime.Now.
Format it as desired and set it as text of the Label control.

Coding

1. Adding a String to ComboBox from TextBox

```
Public Class Form1
   Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
      ComboBox1.Items.Add(TextBox1.Text)
      TextBox1.Clear()
   End Sub
End Class
```

This code adds the text from TextBox1 to ComboBox1 when Button1 is clicked.

2. Displaying Hierarchical Data in TreeView

```
Public Class Form2
   Private Sub Form2_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
      Dim rootNode As TreeNode = New TreeNode("Root")
      Dim childNode As TreeNode = New TreeNode("Child")
      rootNode.Nodes.Add(childNode)
      TreeView1.Nodes.Add(rootNode)
   End Sub
End Class
```

This code creates a simple tree structure with a root node and one child node.

3. Handling User-Defined Exceptions

```
Public Class CustomException
```

```
    Inherits Exception
    Public Sub New(message As String)
        MyBase.New(message)
    End Sub
End Class

Public Class Form3
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
        Try
            Throw New CustomException("This is a custom exception.")
        Catch ex As CustomException
            MessageBox.Show(ex.Message)
        End Try
    End Sub
End Class
```

This example demonstrates how to create and handle a custom exception.

4. Employee Details Using Constructors and Member Functions

```
Public Class Employee
    Public Property Name As String
    Public Property Age As Integer

    Public Sub New(name As String, age As Integer)
        Me.Name = name
        Me.Age = age
    End Sub

    Public Function GetDetails() As String
        Return $"Name: {Name}, Age: {Age}"
    End Function
End Class

Public Class Form4
```

```
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
        Dim emp As New Employee("John Doe", 30)
        MessageBox.Show(emp.GetDetails())
    End Sub
End Class
```

This code defines an Employee class with a constructor and a method to get employee details.

5. Demonstrating Various Events

```
Public Class Form5
    Private Sub Form5_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
        MessageBox.Show("Form Loaded")
    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
        MessageBox.Show("Button Clicked")
    End Sub

    Private Sub Form5_MouseDown(sender As Object, e As MouseEventArgs)
Handles MyBase.MouseDown
        MessageBox.Show("Mouse Down on Form")
    End Sub

    Private Sub Form5_KeyDown(sender As Object, e As KeyEventArgs) Handles
MyBase.KeyDown
        MessageBox.Show($"Key Down: {e.KeyCode}")
    End Sub
End Class
```

This application demonstrates the Click, Mouse Down, Key Down, and Form Load events.

6. File Menu with Menu Items

```
Public Class Form6

    Private Sub NewToolStripMenuItem_Click(sender As Object, e As EventArgs)
Handles NewToolStripMenuItem.Click
        ' Code for New action here.
    End Sub

    ' Similar methods for Open, Save, Print and Exit

End Class

' Add menu items in the designer for File and Edit menus.
```
You would create menu items in the designer and handle their click events accordingly.

7. Student Information Database Operations

```
' Assuming you have a database connection set up.
Public Class StudentDatabaseOperations

    Public Sub AddStudent(name As String)
        ' Code to add student to database.
    End Sub

    Public Sub DeleteStudent(id As Integer)
        ' Code to delete student from database.
    End Sub

    Public Sub UpdateStudent(id As Integer, name As String)
        ' Code to update student information in database.
    End Sub

End Class
```
You would implement the methods to interact with your database accordingly.

8. Web Form to Show Current Date and Time

xml

```
<%@ Page Language="VB" AutoEventWireup="false"
CodeBehind="CurrentDate.aspx.vb" Inherits="YourNamespace.CurrentDate" %>

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title>Current Date</title>
</head>
<body>
   <form id="form1" runat="server">
      <div>
         <asp:Label ID="LabelDateTime" runat="server" />
         <asp:Button ID="ButtonShowDateTime" runat="server" Text="Show
Current Date & Time" OnClick="ButtonShowDateTime_Click" />
      </div>
   </form>
</body>
</html>

' Code-behind (CurrentDate.aspx.vb):
Protected Sub ButtonShowDateTime_Click(sender As Object, e As EventArgs)
   LabelDateTime.Text = DateTime.Now.ToString("F") ' Full date/time pattern
(long date and long time).
End Sub
```