

Microsoft

Exam 70-511

**TS: Windows Applications Development with Microsoft .NET
Framework 4**

Version: 58.5

[Total Questions: 228]

Topic 1, C#**Question No : 1 - (Topic 1)**

You use Microsoft .NET Framework 4 to create a Windows Forms application.

You add a new class named Customer to the application. You select the Customer class to create a new object data source.

You add the following components to a Windows Form:

- ✍ A BindingSource component named customerBindingSource that is data-bound to the Customer object data source.
- ✍ A set of TextBox controls to display and edit the Customer object properties. Each TextBox control is data-bound to a property of the customerBindingSource component.
- ✍ An ErrorProvider component named errorProvider that validates the input values for each TextBox control.

You need to ensure that the input data for each TextBox control is automatically validated by using the ErrorProvider component.

Which two actions should you perform? (Each correct answer presents part of the solution. Choose two.)

- A.** Implement the validation rules inside the Validating event handler of each TextBox control by throwing an exception when the value is invalid.
- B.** Implement the validation rules inside the TextChanged event handler of each TextBox control by throwing an exception when the value is invalid.
- C.** Implement the validation rules inside the setter of each property of the Customer class by throwing an exception when the value is invalid.
- D.** Add the following code segment to the InitializeComponent method of the Windows Form.
`this.errorProvider.DataSource = this.customerBindingSource;`
- E.** Add the following code segment to the InitializeComponent method of the Windows Form.
`this.errorProvider.DataSource = this.customerBindingSource.DataSource;`
`this.errorProvider.DataMember = this.customerBindingSource.DataMember;`

Answer: C,D

Question No : 2 - (Topic 1)

You use Microsoft Visual Studio 2010 and Microsoft .NET Framework 4 to create a Windows Presentation Foundation (WPF) application.

The application contains a ListBox control named `lbxItems` that is data-bound to a collection of objects. Each object has a `DisplayValue` property.

You add a Button control to the application.

You need to ensure that the `Content` property of the Button control is data-bound to the `DisplayValue` property of the selected item of `lbxItems`.

Which binding expression should you use?

- A. `{Binding ElementName= lbxItems, Source=SelectedItem, Path=DisplayValue}`
- B. `{Binding Source=lbxItems, ElementName=SelectedItem, Path=DisplayValue}`
- C. `{Binding ElementName=lbxItems, Path=SelectedItem.DisplayValue}`
- D. `{Binding Source=lbxItems, Path=SelectedItem.DisplayValue}`

Answer: C

Question No : 3 - (Topic 1)

You use Microsoft Visual Studio 2010 and Microsoft .NET Framework 4 to create a Windows Presentation Foundation (WPF) application.

You write the following code fragment.

```
< TextBox Text="{Binding Path=StringMember} " / >
```

You need to ensure that the `StringMember` property of the data-bound object is updated immediately when the user types in the TextBox control.

Which binding expression should you use?

- A. `{ Binding Path=StringMember, Mode=TwoWay }`
- B. `{ Binding Path=StringMember, NotifyOnSourceUpdated=True }`
- C. `{ Binding Path=StringMember, NotifyOnTargetUpdated=True }`
- D. `{ Binding Path=StringMember, UpdateSourceTrigger=PropertyChanged }`

Answer: D

Question No : 4 - (Topic 1)

You use Microsoft .NET Framework 4 to create a Windows Presentation Foundation (WPF) application.

You add a ListBox control to the application. The ListBox control is data-bound to an instance of a custom collection class of the Product objects named ProductList.

The number of items of the data-bound collection is fixed. However, users can modify the properties of each of the Product objects in the collection.

You need to ensure that changes made on the Product objects are automatically reflected in the ListBox control.

What should you do?

- A. Implement the INotifyPropertyChanged interface in the Product class.
- B. Implement the INotifyCollectionChanged interface in the ProductList class.
- C. Set the Mode property of the Binding object of the ListBox control to TwoWay.
- D. Set the UpdateSourceTrigger property of the Binding object of the ListBox control to PropertyChanged.

Answer: A

Question No : 5 - (Topic 1)

You use Microsoft .NET Framework 4 to create a Windows Presentation Foundation (WPF) application.

You plan to allow users to customize the background colors, foreground colors, and font style of the application. You also plan to add a TextBlock control to the application.

You write the following code fragment. (Line numbers are included for reference only.)

```

01 < Window.Resources >
02   < src:ocColorSchemes x:Key="colorschemes" / >
03   < src:ocFonts x:Key="fonts" / >
04   ...
05   < DataTemplate x:Key="StringTemplate" > < TextBlock
06     Text="(Binding Path=.)" / > < /DataTemplate >
07   ...
08   < HierarchicalDataTemplate x:Key="ColorSchemeTemplate"
09     ItemsSource="{Binding Path=ocColors}"
10     ItemTemplate="{StaticResource
11       StringTemplate}" > < TextBlock Text="{Binding
12       Path=.)" / >
13   < /HierarchicalDataTemplate >
14 < /Window.Resources >
15

```

You need to ensure that the ContextMenu control is associated with the TextBlock control. You also need to ensure that the properties that can be customized are shown hierarchically.

Which code fragment should you insert at line 15?

A. < Grid >

< Menu >

< MenuItem Header="Color Scheme" DataContext="{StaticResource colorschemes}"
ItemsSource="{Binding Path=.}" ItemTemplate="{StaticResource StringTemplate}" / >

< MenuItem Header="Font" DataContext="{StaticResource fonts}" ItemsSource="{Binding
Path=.}" ItemTemplate="{StaticResource StringTemplate}" / >

< /Menu >

< TextBlock ttdth="200" Height="100" Background="LightBlue" / > < /Grid >

B. < Window.ContextMenu >

< ContextMenu >

< MenuItem Header="Color Scheme" DataContext="{StaticResource colorschemes}"
ItemsSource="{ Binding Path ItemTemplate="{StaticResource StringTemplate}" / >

< MenuItem Header="Font" DataContext="{ StaticResource fonts!" ItemsSource=,,{
Binding Path=.}" ItemTemplate="{StaticResource StringTemplate}" / >

< /ContextMenu >

< /Window.ContextMenu >

C. < Grid >

< TextBlock Tjidth="200" Height= "100" Background="LightBlue" / > < /Grid >

< Window.ContextMenu >

< ContextMenu >

< TextBlock Width="200" Height="100" Background="LightBlue" / >

< MenuItem Header="Color Scheme" DataContext="{StaticResource colorschemes}"
ItemsSource="{Binding Path*." ItemTemplate="{StaticResource ColorSchemeTemplate}"

```

/>
< MenuItem Header="Font" DataContext="{StaticResource fonts}" ItemsSource="{Binding
Path=." ItemTemplate="{StaticResource StringTemplate}" >< /MenuItem >
< /ContextMenu >
</Window.ContextMenu >
D. < Grid >
< TextBlock Uidth="200" Height="100" Background="LightBlue" >
< TextBlock.ContextMenu >
< ContextMenu >
< MenuItem Header="Color Scheme" DataContext="{StaticResource colorschemes}"
ItemsSource="{Binding Path=.) ItemTemplate="{StaticResource ColorSchemeTemplate}" /
>
< MenuItem Header="Font" DataContext="{StaticResource fonts}" ItemsSource="{Binding
Path=." ItemTemplate="{StaticResource StringTemplate}" > < /MenuItem >
</ContextMenu >
< /TextBlock.ContextMenu >
< /TextBlock > </Grid >

```

Answer: D

Question No : 6 - (Topic 1)

You use Microsoft .NET Framework 4 to create a Windows Presentation Foundation (WPF) application.

You write the following code fragment.

```

<Window x:Class="MyProject.MainWindow" xmlns:local="clr-namespace:MyProject">
  <Window.Resources>
    <local:MyConverter x:Key="myConverter" />
  </Window.Resources>
  <ListBox Name="box">
    <ListBox.ItemTemplate>
      <DataTemplate>
        <TextBlock
          Text="{Binding Converter={StaticResource myConverter}, ConverterParameter=formatText}"
          Background="{Binding Converter={StaticResource myConverter}, ConverterParameter=formatColor}"
          IsEnabled="{Binding Converter={StaticResource myConverter}}" />
      </DataTemplate>
    </ListBox.ItemTemplate>
  </ListBox>
</Window>

```

You need to implement the MyProject.MyConverter class.

What should you do?

- A. Implement the IValueConverter interface.
- B. Implement the IMultiValueConverter interface.
- C. Inherit from the TypeConverter class.
- D. Apply the TypeConverterAttribute attribute.

Answer: A

Question No : 7 - (Topic 1)

You use Microsoft .NET Framework 4 to create a Windows Presentation Foundation (WPF) application. The application contains an ObservableCollection object named Pictures that contains several Picture objects. Each Picture object contains the Name and PictureFilePath properties.

You add a ListBox control to the application.

You need to ensure that the pictures are displayed in the ListBox control.

Which code fragment should you use?

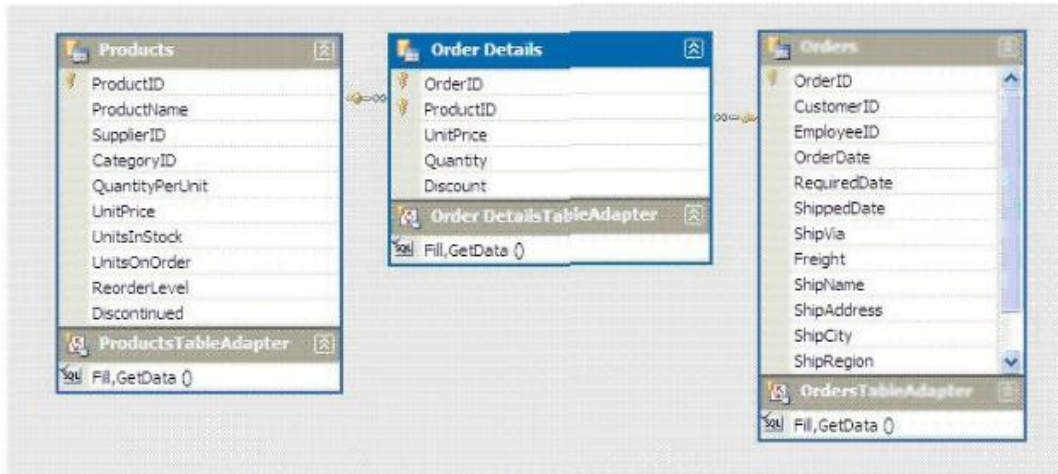
- A. `<ListBox ItemsSource="{Binding Source={StaticResource pictures}}">
<ListBox.ItemTemplate>
<DataTemplate>
<TextBlock>
<TextBlock.Text>
<Binding Path="PictureFilePath" /> </TextBlock.TextX/TextBlock> </DataTemplate>
</ListBox.ItemTemplate> </LiscBox>`
- B. `<ListBox ItemsSource="{Binding Source={StaticResource pictures}}">
<ListBox.ItemTemplate>
<DataTemplate>
<Image Source="{Binding Source={StaticResource pictures}/ Path=PictureFilePath}"/>
</DataTemplate> </ListBox.ItemTemplate> </ListBox>`
- C. `<ListBox ItemsSource="{Binding Source={StaticResource pictures}}">
<ListBox.ItemTemplate>
<DataTemplate>
<TextBlock>
<TextBlock.Text>
<Binding Path="Name" /> </TextBlock.Text></TextBlock> </DataTemplate> </ListBox.ItemTemplate> </LiscBox>`
- D. `<ListBox ItemsSource="{Binding Source={StaticResource pictures}}">
<ListBox.ItemTemplate>
<DataTemplate>
<Image Source="{Binding Path=PictureFilePath}"/> </DataTemplate>
</ListBox.ItemTemplate> </ListBox>`

Answer: D

Question No : 8 - (Topic 1)

You use Microsoft .NET Framework 4 to create a Windows Forms application.

You have a dataset as shown in the following exhibit.



You plan to add a DataGridView to display the dataset.

You need to ensure that the DataGridView meets the following requirements:

- ✎ Shows Order Details for the selected order.
- ✎ Shows only Order Details for items that have UnitPrice greater than 20
- ✎ Sorts Products by ProductName

Which code segment should you use?

- A.** ordersBindingSource.DataSource = productsBindingSource;
ordersBindingSource.DataMember = "FK_Order_Details_Products";
productsBindingSource.Filter = "UnitPrice > 20";
productsBindingSource.Sort = "ProductName";
- B.** productsDataGridView.DataSource = ordersBindingSource;
productsBindingSource.Filter = "UnitPrice > 20";
productsBindingSource.Sort = "ProductName";
- C.** order_DetailsBindingSource.DataSource = ordersBindingSource;
order_DetailsBindingSource.DataMember = "FK_Order_Details_Orders".
order_DetailsBindingSource.Filter = "UnitPrice > 20";
productsBindingSource.Sort="ProductName";
- C D.** order_DetailsDataGridView.DataSource = ordersBindingSource;
order_DetailsBindingSource.Filter = "UnitPrice > 20";
productsBindingSource.Sort="ProductName";

Answer: C

Question No : 9 - (Topic 1)

You are developing a Windows Forms application that contains a DataGridView control. The DataGridView is composed of several fields that capture the customer's name, address, and phone number.

You have been asked to provide data validation in a DataGridView to prevent users from leaving the name field if the name field is empty.

You need to ensure that users cannot tab out of the name field without entering data.

What should you do?

- A.** Validate the name field in the CellValidating event. Set e.Cancel - true if the name field is empty.
- B.** Validate the name field in the CellErrorTextChanged event. Get e.RowIndex if the name field is empty.
- C.** Validate the name field in the CellEnter event. Set the focus on the name field if the name field is empty.
- D.** Validate the name field in the CancelRowEdit event. Set the focus on the name field if the name field is empty.

Answer: A

Question No : 10 - (Topic 1)

You use Microsoft .NET Framework 4 to create a Windows Presentation Foundation (WPF) application. You write the following code fragment. (Line numbers are included for reference only.)

```
01 <Canvas>
```

```
02 <Rectangle Stroke="Red" Stroke Thickness="5" Height="60"
```

```
03 Width="60" Canvas. Leftoff Canvas.Top-"100"> 04
```

```
05 </Rectangle>
```

06 </Canvas>

You need to rotate the rectangle by 45 degrees by using its upper-left corner as the axis.

Which code fragment should you insert at line 04?

- A. <Rectangle.RenderTransformo>
<RotateTransform Angle="45" CenterX="0" CenterY="0" />
</Rectangle.RenderTransform>
- B. <Rectangle.RenderTransform>
<RotateTransform Angle="45" CenterX="100" CenterY="100" />
</Rectangle.RenderTransform>
- C. <Rectangle.LayoutTransform>
<RotateTransform Angle="45" CenterX="0" CenterY="0"/>
</Rectangle.LayoutTransform>
- D. <Rectangle.LayoutTransform>
<RotateTransformo Angle="45" CenterX="100" CenterY="100"/>
</Rectangle.LayoutTransform>

Answer: A

Question No : 11 - (Topic 1)

You are developing a Windows Presentation Foundation (WPF) application.

You have the following style defined in the app.xaml file.

```
<Style x:Key="btnItalic" x:Name="styleItalic"
TargetType="{x:Type Button}">
<Setter Property="FontStyle" Value="Italic"/> </Style>
```

You need to apply this style to a button named button1 in the application at run time.

Which code segment should you use?

- A. button1.Style =
this.FindName("styleItalic") as Style;
- B. button1.Style =