



The Top Certification Site **OVER 1000 EXAMS FROM ALL VENDORS**

- Verified Answers and Explanations
- Printable questions and answers
- Update per 15-20 Days
- Instant Download
- Security Multi Order
- 24*7 Support
- Pass on Your First Try Guarantee

IT Exam World .com

 **interactive Exams**
Self Exam Engine

 **Questions & Answers**
With Explanations

 **Study Guides**

 **Preparation Labs**

 **Audio Exams**

Exam Code: 070-526
Microsoft .NET Framework 2.0 -
Windows -Based Client Development

Demo Version

To Access Full Version, Please go to
www.itexamworld.com

QUESTION 1:

You are creating a Windows Form that contains several ToolStrip controls. You need to add functionality that allows a user to drag any ToolStrip control from one edge of the form to another. What should you do?

- A. Configure a ToolStripContainer control to fill the form. Add the ToolStrip controls to the ToolStripContainer control.
- B. Configure a Panel control to fill the form. Set the Anchor properties of the ToolStrip controls to Top, Bottom, Left, Right.
- C. Add a ToolStrip controls to another ToolStrip control that is hosted by a ToolStripControlHost control.
- D. Add the ToolStrip controls to the form. Set the Anchor properties of the ToolStrip controls to Top, Bottom, Left, Right. Set the FormBorderStyle property of the form to SizableToolWindow.

Answer: A

QUESTION 2:

You need to create a Windows Forms application that uses a nonrectangular form as its user interface. What should you do?

- A. Set the FormBorderStyle property of the form to None. Set the BackgroundImage property of the form to a bitmap file that represents the shape you want form to take. Set the TransparencyKey property to the background color of the bitmap file.
- B. Set the FormBorderStyle property of the form to None. Set the BackgroundImage property of the form to a bitmap file that represents the shape you want the form to take. Set the TransparencyKey property to Transparent.
- C. Set the FormBorderStyle and BackgroundImageLayout properties to None. Set the BackgroundImage property of the form to a bitmap file that represents the shape

you want the form to take.
Set the TransparencyKey property to Transparent.

D. Set the FormBorderStyle property to None and the BackColor property to Control.
Set the BackgroundImage property of the form to a bitmap file that represents the shape you want the form to take.
Set the TransparencyKey property to Transparent.

Answer: A

QUESTION 3:

You create a Windows Forms application. Your application executes a background thread.

You need to construct the thread to exit, but you also need to notify the main thread when the background thread has ended.

What should you do?

- A. Call the Abort method of the thread.
- B. Call the Interrupt method of the thread.
- C. Call the Join method of the thread.
- D. Call the Sleep method of the thread.

Answer: C

QUESTION 4:

You are creating a Windows Forms application that uses a drag-and-drop operation to enable users to copy customer data between a ListBox control and RichTextBox control. The ListBox displays a list of customer Ids to the user. Each item in the ListBox is associated with a custom external data type named CustomerData. The data type stores the customer name along with other customer information, including the address and postal code. You need to ensure that when the user drags a customer name from the ListBox to the RichTextBox all of the information in your custom data type is moved into the RichTextBox. What should you do?

- A. Initiate the drag-and-drop operation in the MouseDown event for the ListBox. Call the DoDragDrop method for the ListBox, passing in an instance of CustomerData. Use the GetFormats method in the DragEnter event for the RichTextBox to access the custom data type.
- B. Initiate the drag-and-drop operation in the MouseDown event for the ListBox. Call the DoDragDrop method for the ListBox, passing in an instance of CustomerData.

Use the GetData method in the DragDrop event for the RichTextBox to access the custom data type.

C. Initiate the drag-and-drop operation in the MouseDown event for the ListBox. In the DragEnter or DragDrop events for the RichTextBox, set the Effect property to DragDropEffects.All. Use the GetFormats method in the DragEnter event for the RichTextBox to access the custom data type.

D. Initiate the drag-and-drop operation in the MouseDown event for the ListBox. In the DragEnter or DragDrop events for the RichTextBox, set the Effect property to DragDropEffects.All. Use the GetDataPresent method in the DragEnter event for the RichTextBox to access the custom data type.

QUESTION 5:

You are creating multiple-document interface (MDI) Windows Forms application. You need to configure the main form to function as the parent form and a second form to function as the child form. What should you do?

- A. Set the IsMdiContainer property of the parent form to True. Set the MdiParent property of the child form to the parent form.
- B. Set the IsMdiContainer property of the parent form to True. Set the Parent property of the child form to the parent form.
- C. Add the child form to the Controls collection of the parent form. Set the MdiParent property of the child form to the parent form.
- D. Add the child form to the Controls collection of the parent form. Set the Parent property of the child form to the parent form.

Answer: A

QUESTION 6:

You are adding accessibility functionality to a custom, owner-drawn control named Legend. You create a class named AccessibleLegend, which is derived from the AccessibleObject class. The Legend control overrides the GetChild method and returns an AccessibleLegend object. You need to ensure that when the Legend control is disabled, the control still returns the appropriate value for the State property of the AccessibleLegend object. Which value should you configure the State property to return?

- A. AccessibleStates.Invisible
- B. AccessibleStates.Protected

-
- C. AccessibleStates.ReadOnly
 - D. AccessibleStates.Unavailable

Answer: D

QUESTION 7:

You are creating a Windows Forms application. Your application uses a custom control. The Custom control is based on a standard button control. You add several extra properties to the control. Some of these properties are read-only. You need to ensure that the read-only properties are not displayed in the Properties window of the design environment.

- A. Decorate the read-only properties by using the EditorBrowsable attribute, and then set the EditorBrowsable attribute to EditorBrowsableState.Never.
- B. Use the DesignTimeVisible attribute, and then set the DesignTimeVisible attribute to False.
- C. Decorate the read-only properties by using the Browsable attribute, and then set the Browsable attribute to False.
- D. Decorate the read-only properties by using the DisplayName attribute, and then set the DisplayName attribute to null.

Answer: C

QUESTION 8:

You create a custom control by extending a standard TextBox control. The custom control adds a new property called ValidationColor. You need to ensure that users can select the color for the ValidationColor property from the Properties window at design time by using the color action palette that is available for other standard Windows Forms controls. What should you do?

- A. Create a custom context menu that contains the chosen color palette and logic. Implement the IContainerControl interface for your control.
- B. Create a custom dialog box that contains the chosen color palette and logic. Configure the FormBorderStyle property of the dialog box to None. Implement the IContainerControl interface for your control, and use its ActivateControl method to activate the dialog box.
- C. Create a custom dialog box that contains the chosen color palette and logic. Configure the FormBorderStyle property of the dialog box to None. In the Set method of your property, write code to instantiate the dialog box and return the selected value.

D. Declare the type of the property as System.Drawing.Color.

Answer: D

QUESTION 9:

You created a custom Windows Forms control that contains width, Height, and SquareFootage properties. The SquareFootage property contains the multiplied value of Width and Height.

You need to make SquareFootage visible, while disabling it in the property grid. What should you do?

- A. Apply the NotSerialized() attribute to the SquareFootage property.
- B. Apply the DesignerSerializationVisibility() attribute to the SquareFootage property, and pass in a value of DesignerSerializationVisibility.Content as a parameter.
- C. Apply the EditorBrowsable attribute to the SquareFootage property, passing in a value of EditorBrowsableState.Never as a parameter.
- D. Implement the property by using only a Get accessor.

Answer: D

QUESTION 10:

You are customizing a Windows Form. You want to display a custom icon for your composite control named DialerControl in the toolbox.

You need to customize the toolbox icon. You want to achieve this goal by using the minimum amount of effort.

Which action or actions should you perform? (Choose all that apply.)

- A. Add a bitmap named DialerControl.bmp to the user control project.
- B. Change the build action of the bitmap to Embedded Resource.
- C. Add a ToolboxBitmap attribute to the DialerControl class, passing in DialerControl.bmp as a parameter.
- D. Add a ToolboxBitmap attribute to the DialerControl class, passing in typeof(DialerControl) as a parameter.

Answer:

A. B