Microsoft 70-526 TS: MS.NET Framework 2.0-Windows Based Client Development

Practice Test

Version 2.2

QUESTION NO: 1

You are creating a Windows Forms application. You add an ErrorProvider component named erpErrors and a DateTimePicker control named dtpStartDate to the application. The application also contains other controls. You need to configure the application to display an error notification icon next to dtpStartDate when the user enters a date that is greater than today's date. Which two actions should you perform? (Each correct answer presents part of the solution. Choose two.)

A. For the Validated event of dtpStartDate, create an event handler named VerifyStartDate.
B. In VerifyStartDate, call erpErrors.SetError(dtpStartDate, "Date out of range") if the value of dtpStartDate.Value is greater than today's date.
C. In the Properties Window for dtpStartDate, set the value of Error on erpErrors to Date out of range.
D. In VerifyStartDate, call erpErrors.SetError(dtpStartDate, null) if the dtpStartDate.Value is greater than today's date.
E. For the Validating event of dtpStartDate, create an event handler named VerifyStartDate.

Answer: D,E

QUESTION NO: 2

You are customizing a Windows Form to asynchronously update a database. You need to ensure that the form displays a message box to the user that indicates the success or failure of the update. Which three code segments should you use? (Each correct answer presents part of the solution. Choose three.)

A. void WorkHandler(object sender, DoWorkEventArgs e) {
//...
e.Result = true; }
B. void WorkHandler(object sender, DoWorkEventArgs e) {
//...
bgwExecute.ReportProgress(100, true);
}
C. void ProgressHandler(object sender, ProgressChangedEventArgs e) {
bool result = (bool)e.UserState;
MessageBox.Show("Update " + (result ? "was successful" : "failed"));
}
D. void CompletedHandler(object sender, RunWorkerCompletedEventArgs e) {
bool result = (bool)e.Result;
MessageBox.Show("Update " + (result ? "was successful" : "failed"));
}
E. private void StartBackgroundProcess() {
bgwExecute.ProgressChanged +=

```
new ProgressChangedEventHandler(CompletedHandler);
ThreadStart tsBackground = new ThreadStart(WorkHandler);
bgwExecute.RunWorkerAsync(tsBackground);
}
F. private void StartBackgroundProcess() {
bgwExecute.RunWorkerCompleted +=
new RunWorkerCompletedEventHandler(CompletedHandler);
ThreadStart tsBackground = new ThreadStart(WorkHandler);
bgwExecute.RunWorkerAsync(tsBackground);
}
G. private void StartBackgroundProcess() {
bgwExecute.DoWork += new DoWorkEventHandler(WorkHandler);
bgwExecute.RunWorkerCompleted +=
new RunWorkerCompletedEventHandler(CompletedHandler);
bgwExecute.RunWorkerAsync();
}
```

Answer: A,D,G

QUESTION NO: 3

You are creating a custom control that displays an image in the background. You notice that when the control is resized, the background image flickers while the control is repeatedly repainted. You need to eliminate the background image flicker. Which three code segments should you use? (Each correct answer presents part of the solution. Choose three.)

A. this->SetStyle(ControlStyles::UserPaint, true);
B. this->SetStyle(ControlStyles::OptimizedDoubleBuffer, true);
C. this->SetStyle(ControlStyles::Opaque, true);
D. this->SetStyle(ControlStyles::ResizeRedraw, true);
E. this->SetStyle(ControlStyles::AllPaintingInWmPaint, true);

Answer: A,B,D

QUESTION NO: 4

You are customizing a Windows Form. You need to add an input control that provides AutoComplete suggestions to the user as the user types. Which two controls can you use to achieve this goal? (Each correct answer presents a complete solution. Choose two.)

A. TextBox control set to MultiLine mode

B. RichTextBox control

C. MaskedTextBox control

D. ComboBox control

E. TextBox control set to SingleLine mode

Answer: D,E

QUESTION NO: 5

You are creating a Windows Forms application that prints reports. The application uses the PrintDocument control to print the report and the PrintPreviewDialog control to preview reports as shown in the following code segment.

```
streamToPrint = New StreamReader("FileToPrint.txt")
Try
Dim pd As New PrintDocument()
AddHandler pd.PrintPage, AddressOf pd_PrintPage
Dim ppd As New PrintPreviewDialog()
ppd.Document = pd
ppd.ShowDialog()
Finally
streamToPrint.Close()
End Try
```

When a report is printed by using the Print method on the PrintDocument class, the output is correct. When the report is previewed by using the Print Preview dialog box, the output is correct. However, when the report is printed by using the Print button in the Print Preview dialog box, a single blank page is produced. You need to ensure that the output is correct when the Print button in the Print Preview dialog box is used. What should you do?

A. In the event handler for the pd.PrintPage event, set the position of the streamToPrint.BaseStream property to 0.

B. In the event handler for the ppd.Click event, set the position of the streamToPrint.BaseStream property to 0.

C. In the event handler for the pd.BeginPrint event, set the position of the streamToPrint.BaseStream property to 0.

D. In the event handler for the ppd.PrintPreviewControl.Click event, set the position of the streamToPrint.BaseStream property to 0.

Answer: C

QUESTION NO: 6

You are creating a Windows Form. You add a TableLayoutPanel control named pnlLayout to the form. You set the properties of pnlLayout so that it will resize with the form. You need to create a three-column layout that has fixed left and right columns. The fixed columns must each remain 50 pixels wide when the form is resized. The middle column must fill the remainder of the form width when the form is resized. You add the three columns in the designer. Which code segment should you use to format the columns at run time?

A. pnlLayout->ColumnStyles[0]->Width = 50;pnlLayout->ColumnStyles[0]->SizeType = SizeType::Absolute;pnlLayout->ColumnStyles[2]->Width = 50;pnlLayout->ColumnStyles[2]->SizeType = SizeType::Absolute;
B. pnlLayout->ColumnStyles->Clear();pnlLayout->ColumnStyles->Add(gcnew ColumnStyle(SizeType::Absolute,50));pnlLayout->ColumnStyles->Add(gcnew ColumnStyle(SizeType::Percent, 100));pnlLayout->ColumnStyles->Add(gcnew ColumnStyle(SizeType::Absolute, 50));
C. pnlLayout->ColumnStyles->Clear();pnlLayout->ColumnStyles->Add(gcnew ColumnStyle(SizeType::Absolute, 50));pnlLayout->ColumnStyles->Add(gcnew ColumnStyle(SizeType::AutoSize, 100));pnlLayout->ColumnStyles->Add(gcnew ColumnStyle(SizeType::Absolute, 50));
D. pnlLayout->ColumnStyles[0]->Width = 50;pnlLayout->ColumnStyles[0]->SizeType = SizeType::Absolute;pnlLayout->ColumnStyles[1]->Width = 100;pnlLayout->ColumnStyles[1]->SizeType = SizeType::AutoSize;pnlLayout->ColumnStyles[2]->Width = 50;pnlLayout->ColumnStyles[2]->SizeType = SizeType::Absolute;

Answer: B

QUESTION NO: 7

A method in your Windows Forms application executes a stored procedure in a Microsoft SQL Server 2005 database, and then executes a second stored procedure in a second SQL Server 2005 database. You need to ensure that the call to the first stored procedure writes changes only if the call to the second stored procedure succeeds. Installation requirements prohibit you from introducing new components that use the COM+ hosting model. What should you do?

A. Implement a transactional serviced component.Add methods to this component to encapsulate the connect operation and execution of each stored procedure.Register and use this serviced component.
B. Add a TransactionScope block.Connect to each database and execute each stored procedure within the TransactionScope block.Call the TransactionScope.Complete method if the call to both stored procedure succeeds.
C. Connect to both databases.Call the SqlConnection.BeginTransaction method for each connection.Call the SqlTransaction.Commit method on both returned transactions only if both stored procedures succeed.

D. Add a TryCatchFinally block.Connect to each database and execute each stored procedure in the try block.

Answer: B

QUESTION NO: 8

You are creating a Windows Forms application that targets Microsoft Windows 2000 or later. You need to configure your Setup application for conditional installation. Which part or parts of a Setup application can be configured for conditional installation based on the operating system? (Choose all that apply.)

A. primary output of a project
B. custom actions
C. registry keys and values
D. user interface dialog boxes
E. file type associations

Answer: C,E

QUESTION NO: 9

You are creating a Windows Forms setup application. The default user interface does not meet your needs. You want to provide an additional dialog box that includes two check boxes during the install process. You want the check boxes to give users the option to install two large Help files named Help1 and Help2 during the installation process. You need to customize the interface to meet your needs. Which three actions should you perform? (Each correct answer presents part of the solution. Choose three.)

A. In the Properties window for your setup project, set the PreBuildEvent property to call a command line to display the dialog box.
B. Use the Custom Actions Editor to add the dialog box executable for the setup application to the Install node.
C. In the User Interface Editor, add a Checkboxes dialog box to the Start node of the user interface tree.
D. Set the Checkbox3Visible and Checkbox4Visible properties of the Checkboxes dialog box to False.
E. Create the dialog box and the logic for the dialog box in a separate project.Compile the project into an executable, and add the executable to the setup project.
F. In the File System Editor, set the Condition property for Help1 to the value of the Checkbox1Property property.Set the Condition property for Help2 to the value of the Checkbox2Property property.

Answer: C,D,F

QUESTION NO: 10

You are creating a data-entry application that uses many TextBox controls. Each control requires special validation logic. You need to create an extended version of the standard Windows Forms TextBox control that has the validation logic built into the control. You start by creating a new Windows Control Library project. What should you do next?

A. In the code, change the class to derive from the UserControl class, and add properties and methods as required.
B. Add a TextBox control to the control design surface, and add properties and methods as required.
C. In the code, change the class to derive from the TextBox class, and add properties and methods as required.
D. In the code, change the class to derive from the Control base class, and add properties and methods as required.

Answer: C

QUESTION NO: 11

You created a custom action for your Windows setup application. The custom action runs a standard Console application at the end of the installation process. You place the custom action in the Install node of the Custom Actions tree in the Custom Actions Editor. The Console application executable performs correctly when run as a stand-alone application. However, when you run the Microsoft Windows Installer package that was created by your setup application, the custom action does not run. Everything else works fine. You need to ensure that the console application runs during the install. What should you do?

A. Set the InstallerClass property of your custom action to False.
B. Place the custom action in the Commit node of the Custom Actions tree rather than in the Install node.
C. Set the DetectNewerInstalledVersion property for your setup application to True.
D. Set the InstallAllUsers property for your setup application to True.

Answer: A

QUESTION NO: 12

You are creating a Windows Forms application. You create a System.Threading.Semaphore instance in the application by using the following code.

Semaphore^ threadPool = gcnew Semaphore(0, 3)
;//...
threadPool->Release(4);
What does this code do?

A. Throws a SemaphoreFullException.
B. Sets the internal counter for the semaphore to 4.
C. Sets the internal counter for the semaphore to 0.
D. Throws a ThreadAbortException.

Answer: A

QUESTION NO: 13

You are creating a Windows Forms application that implements a master/detail form by using two DataGridView controls. You populate a dataset with a master table and a details table. You set the DataSource property of the master DataGridView control to the dataset. You set the DataMember property to the name of the master table. You also set the DataSource property of the details DataGridView control to the dataset. You need to ensure that the details DataGridView control displays only the child rows of the selected master row. What should you do?

A. Define a data relation between the master table and details table in the dataset.Bind the details DataGridView control to the dataset.Set the DataMember property of the child DataGridView control to the name of the details table.
B. Add a foreign key constraint to the dataset.Set the DataMember property of the child DataGridView control to the name of the details table.
C. Add a foreign key constraint to the dataset.Set the DataMember property of the child DataGridView control to the name of the foreign key constraint.
D. Define a data relation between the master table and details table in the dataset.Set the DataMember property of the child DataGridView to the name of the data relation.

Answer: D

QUESTION NO: 14

You create a Windows user control named EmployeeData. You want to provide design-time layout support for EmployeeData so that it will easily align with other controls on the form. You create a class named SnapLineDesigner that inherits from the System.Windows.Forms.Design.ControlDesigner class. You add a Designer attribute to the EmployeeData class and pass in SnapLineDesigner as the designerTypeName parameter. You need to provide the design-time layout support for EmployeeData. What should you do?

A. In the constructor of SnapLineDesigner, add items to the SnapLines collection of SnapLineDesigner.
B. In the constructor of SnapLineDesigner, add items to the SnapLines collection of EmployeeData.
C. In the constructor of EmployeeData, add items to the SnapLines collection of SnapLineDesigner.
D. In the constructor of EmployeeData, add items to the SnapLines collection of EmployeeData.

Answer: A

QUESTION NO: 15

You are creating a Windows Forms application. You add an ErrorProvider component named erpErrors and a DateTimePicker control named dtpStartDate to the application. The application also contains other controls. You need to configure the application to display an error notification icon next to dtpStartDate when the user enters a date that is greater than today's date. Which two actions should you perform? (Each correct answer presents part of the solution. Choose two.)

A. In VerifyStartDate, call erpErrors.SetError(dtpStartDate, "Date out of range") if the value of dtpStartDate.Value is greater than today's date.
B. In the Properties Window for dtpStartDate, set the value of Error on erpErrors to Date out of range.
C. In VerifyStartDate, call erpErrors.SetError(dtpStartDate, null) if the dtpStartDate.Value is greater than today's date.
D. For the Validated event of dtpStartDate, create an event handler named VerifyStartDate.
E. For the Validating event of dtpStartDate, create an event handler named VerifyStartDate.

Answer: C,E

QUESTION NO: 16

You are creating a Windows Forms application. You implement the asynchronous design pattern by using a delegate named DataUpdateHandler. DataUpdateHandler executes a method named DataUpdater. You create an instance of the DataUpdateHandler that has the variable name Updater. DataUpdater uses a DataSet class as a parameter and returns an integer value that indicates the number of rows that have been updated. You need to use a method named DisplayResults to display the results when the database update is complete. Which code segment should you use?

A. Dim result As IAsyncResult = Updater.BeginInvoke(mData, Nothing, _
Nothing)
'...
Dim value As Integer = result.AsyncState
DisplayResults(value)
B. Dim result As IAsyncResult = Updater.BeginInvoke(mData, Nothing, _
Nothing)
'...
Dim value As Integer = Updater.EndInvoke(result)
DisplayResults(value)
C. Dim result As IAsyncResult = Updater.BeginInvoke(mData, Nothing, _ "DataUpdater")
'...
Updater.EndInvoke(result)
D. Dim result As IAsyncResult = Updater.BeginInvoke(mData, Nothing, _
"DisplayResults")
'...
Updater.EndInvoke(result)

Answer: B

QUESTION NO: 17

You are creating a Windows Forms application. The application executes a stored procedure that takes several seconds to complete. The stored procedure is invoked to populate a SqlDataReader object. You need to ensure that the application remains responsive to the user while the stored procedure is executing. What should you do?

A. Set the Notification property of a SqlCommand object to a SqlNotificationRequest object. Call the SqlCommand.ExecuteReader method.Gather results on a background thread.
B. Create and bind a SqlDependency object to a SqlCommand object. Call the SqlCommand.ExecuteReader method.Associate an OnChanged event handler with the SqlDependency object. Gather results in the OnChanged event handler method.
C. Use the SqlCommand.BeginExecuteReader method to call the stored procedure.Retrieve results by using the EndExecuteReader method.
D. Use the SqlCommand.ExecuteReader method.Set the behavior parameter of this method to CommandBehavior.SequentialAccess.

Answer: C

QUESTION NO: 18