

# The best learning assistants

**Exam : Microsoft 70-306**

**Title :** Developing and Implementing  
Windows-based Applications

**Update :** Demo



5. You use Visual Studio .NET to create a Windows-based application. On the main application form, you create a TextBox control named textConnectionString. Users can enter a database connection string in this box to access customized data from any database in your company. You also create a Help file to assist users in creating connection strings. The Help file will reside on your company intranet. Your application must load the Help file in a new browser window when the user presses the F1 key, but only if textConnectionString has focus. You must create this functionality by using the minimum amount of code. In which event should you write the code to display the Help file?

- A. textConnectionString\_KeyPress
- B. textConnectionString\_KeyDown
- C. textConnectionString\_KeyUp
- D. textConnectionString\_GiveFeedback
- E. textConnectionString\_HelpRequested

Answer: E

6. You develop a Windows-based application that accesses a Microsoft SQL Server database. The application includes a form named CustomerForm, which contains a Button control named sortButton. The database includes a table named Customers.

Data from Customers will be displayed on CustomerForm by means of a DataGrid control named DataGrid1. The following code segment is used to fill DataGrid1:

```
Private Sub FillDataGrid()  
  
    Dim oConn As New SqlConnection(MyConString)  
  
    Dim oDA As New SqlDataAdapter(_  
        "SELECT CustomerID, CompanyName, ContactName, " & _  
        "Phone FROM Customers". oConn)  
  
    Dim oDS As New DataSet()  
  
    oDA.MissingSchemaAction = MissingSchemaAction.AddWithKey  
  
    oDA.Fill(oDS, "Customers")  
  
    Dim oDV As New DataView(oDS.Tables("Customers"))  
  
    oDV.Sort = "CustomerName ASC, ContactName ASC"  
  
    oDV.ApplyDefaultSort = True  
  
    DataGrid1.DataSource = oDV  
  
End Sub
```

The primary key for Customers is the CustomerID column. You must ensure that the data will be displayed in ascending order by primary key when the user selects sortButton.

What should you do?

- A. Set the Sort property of the DataView object to an empty string.
- B. Set the ApplyDefaultSort property of the DataView object to False.
- C. Include an ORDER BY clause in the SELECT statement when you create the DataAdapter object.
- D. Set the RowFilter property of the DataView object to CustomerID.

Answer: A

7. You develop a kiosk application that enables users to register for an e-mail account in your domain. Your application contains two TextBox controls named `textName` and `textEmail`. Your application is designed to supply the value of `textEmail` automatically. When a user enters a name in `textName`, an e-mail address is automatically assigned and entered in `textEmail`. The `ReadOnly` property of `textEmail` is set to `True`. Your database will store each user's name. It can hold a maximum of 100 characters for each name. However, the database can hold a maximum of only 34 characters for each e-mail address. This limitation allows 14 characters for your domain, `@proseware.com`, and 20 additional characters for the user's name. If a user enters a name longer than 20 characters, the resulting e-mail address will contain more characters than the database allows. You cannot make any changes to the database schema. You enter the following code segment in the `Leave` event handler of `textName`: `textEmail.Text = textName.Replace(" ", ".") _ & "@proseware.com"` Now you must ensure that the automatic e-mail address is no longer than 34 characters. You want to accomplish this goal by writing the minimum amount of code and without affecting other fields in the database. What should you do?

- A. Set the `textName.Size` property to "1,20".
- B. Set `textEmail.Size` property to "1,34".
- C. Set `textName.AutoSize` property to `True`.
- D. Set `textEmail.AutoSize` property to `True`.
- E. Set `textName.MaxLength` property to 20.
- F. Set `textEmail.MaxLength` property to 34.
- G. Change the code in `textName_Leave` to ensure that only the first 20 characters of `textName.Text` are used.
- H. Use an `ErrorProvider` control to prompt for a revision if a user enters a name longer than 20 characters.

Answer: G

8. You use Visual Studio .NET to create an application that will track vehicles leased by your company. You design object classes to model the vehicles by using the following code segment:

```
Class Vehicle
```

```
Public Sub Lease()
```

```
'Execute code to lease a vehicle.
```

```
End Sub
```

```
End Class
```

```
Class Truck
```

```
Inherits Vehicle
```

```
Public Shadows Sub Lease()
```

```
MyBase.Lease()
```

```
'Execute code to lease a truck.
```

```
End Sub
```

```
End Class
```

Your application defines the following code segment in the Load event handler for the form:

```
Dim newTruck As New Truck()
```

```
newTruck.Lease()
```

Your application defines the following code segment in the Load event handler for the form:

```
Dim newTruck As New Truck()
```

```
newTruck.Lease()
```

What happens when you try to run the application?

- A. A compilation error occurs.
- B. Only the code in the Truck.Lease method executes.
- C. Only the code in the Vehicle.Lease method executes.
- D. The code in the Vehicle.Lease and Truck.Lease methods executes.

Answer: D

9. You develop a Windows-based order entry application. The application uses a DataSet object named customerDataSet to manage client data while users edit and update customer records. The DataSet object contains a DataTable object named creditAuthorizationDataTable. This object contains a field named MaxCredit, which specifies the credit limit for each customer. Customer credit limits cannot exceed \$20,000. Therefore, your application must verify that users do not enter a larger amount in MaxCredit. If a user does so, the user must be notified, and the error must be corrected, before any changes are stored permanently in the database. You create a procedure named OnRowChanged. This procedure sets the DataTable.Row.RowError property for each row that includes an incorrect credit limit. OnRowChanged will run each time a user changes data in a row. OnRowChanged contains the following code segment:

```
Sub OnRowChanged(sender As Object, args As _ &.  
DataRowChangeEventArgs) Handles _ &.  
creditAuthorizationDataTable.RowChanged &.  
If Cdbl(args.Row("MaxCredit")) > 20000 Then &.  
args.Row.RowError = "Over credit limit."  
End If  
End Sub
```

Before updating the database, your application must identify and correct all rows that are marked as having an error. Which code segment should you use?

- A. For Each myRow In \_ creditAuthorization  
    DataTable.GetErrors()  
    MessageBox.Show("CustID = " & \_  
        myRow("CustID").ToString() & " " & \_  
        "Error = " & myRow.RowError)  
Next
- B. For Each myRow In \_  
    CreditAuthorization  
    DataTable.HasErrors()  
    MessageBox.Show("CustID = " & \_  
        myRow("CustID").ToString() & " " & \_  
        "Error = " & myRow.RowError)  
Next
- C. Dim vw As DataView = New DataView( \_ customerDataSet.Tables("CreditAuthorization"), \_  
    DataViewRowState.ModifiedCurrent) MessageBox.Show("CustID = " & vw("CustID") \_ .ToString() & "  
Error")
- D. Dim vw As DataView = New DataView( \_ customerDataSet.Tables("CreditAuthorization"), \_  
    DataViewRowState.ModifiedOriginal)  
    MessageBox.Show("CustID = " & vw("CustID") \_  
        .ToString() & " Error")

Answer: A

10. You are developing a Windows-based application that logs hours worked by your employees. Your design goals require you to maximize application performance and minimize impact on server resources. You need to implement a SqlCommand object that will send a SQL INSERT action query to a database each time a user makes a new entry.

To create a function named LineltemInsert, you write the following code. (Line numbers are included for

```
01  Function LineItemInsert(ByVal empID As Integer, _
02  ByVal projectID As Integer, ByVal hrs As Decimal, _
03  ByVal cnn As SqlConnection) As Integer
04  Dim SQL As String
05  Dim Ret As Integer
06  SQL = "INSERT INTO TimeEntries " & _
07  "(EmpID, ProjectID, Hours) VALUES " & _
08  "(" & empID & ", " & projectID & ", " & hrs & ")"
09  Dim cmd As New SqlCommand(SQL, cnn)
10
11  ' Insert new code.
reference only.)
12
13  End Function
```

Your code must execute the SQL INSERT action query and verify the number of database records that are affected by the query. Which code segment should you add on line 11?

- A. `cnn.Open()`  
`Ret = cmd.ExecuteNonQuery()`  
`cnn.Close()`  
`Return Ret`
- B. `cnn.Open()`  
`Ret = cmd.ExecuteScalar()`  
`cnn.Close()`  
`Return Ret`
- C. `Dim reader As SqlDataReader`  
`cnn.Open()`  
`reader = cmd.ExecuteReader()`  
`cnn.Close()`  
`Return reader.RecordsAffected`
- D. `Dim reader As SqlDataReader`  
`cnn.Open()`  
`reader = cmd.ExecuteReader()`  
`cnn.Close()`  
`Return reader.GetValue()`

Answer: A

11. Your development team is creating an accounting application for your company. Your team uses

Visual Studio .NET to write code, and it stores new code immediately by using Microsoft Visual SourceSafe. You develop a component that uses classes created by another developer on your team. You need to perform unit testing on your component to ensure that it functions properly. However, the classes used by your component are checked out by another developer and are incomplete. You need to perform unit testing as soon as possible. You must not cause any loss of time or resources to your development team. What should you do?

- A. Get the latest version of the classes used by your component. Write new code in the classes to ensure that they behave in the expected manner. Perform unit testing of the component against your revised version of the classes.
- B. Get the latest version of the classes used by your component. Begin your unit test. When you encounter areas of the called classes that are not functioning properly, comment out that portion of the code in your component. Complete the unit test when the classes are checked in.
- C. Create classes that simulate the classes used by your component. Ensure that the simulated classes accept and return parameters that match those of the current version of the classes.
- D. Wait until the classes used by your component are completed. Perform your unit test with the completed classes.

Answer: C

12. You use Visual Studio .NET to create an application that will process telephone orders. Your application uses an XML Web service for credit card verification. It also uses several Microsoft SQL Server stored procedures. The XML Web service contacts a third-party credit card processing service to authorize credit card charges. You have access to the source code for the XML Web service and the stored procedures, which were created by your company and are used by several other applications. Before you deliver your application for comprehensive testing, you need to perform unit testing to ensure that the application performs as expected. Which three actions should you include in your unit test plan? (Each correct answer presents part of the solution. Choose three.)

- A. Use the T-SQL Debugger to step through each stored procedure used by your application and ensure that each one is functioning properly.
- B. Add the source code for the XML Web service to your Visual Studio .NET solution. Use the Visual Studio debugger to step through calls to this code.
- C. Step through the source code for your application and ensure that it is functioning properly given various application parameters.
- D. Examine and validate the input parameters, output parameters, and results from each call to the stored procedures.
- E. Examine and validate the values sent to and received from the XML Web service.
- F. Contact the third-party credit card processing service to verify that credit card charges issued by your application are being delivered properly by the XML Web service.

Answer: CDE

13. You use Visual Studio .NET to create a Windows-based accounting application. Your application will be deployed on computers running Windows 98, Windows 2000 Professional, and Windows XP

Professional. You must ensure that your application records information about errors raised from calls to a procedure named UpdateData. To do so, you plan to implement logging features. Which code segment should you use?

A. Dim oLog As New EventLog()

Dim oType As EventLogEntryType

Try

UpdateData()

Catch oEx As Exception

If Not EventLog.SourceExists("AccountingApp")

Then EventLog.CreateEventSource("AccountingApp", \_ "Application")

End If

oLog.Source = "AccountingApp"

oLog.Log = "Application"

oType = EventLogEntryType.Error

oLog.WriteEntry("Error details: " & \_ oEx.Message, oType)

End Try

B. Dim IFH As Long

Try

UpdateData()

Catch oEx As Exception IFH = FreeFile()

Open logfilePath For Append As #IFH

Print #IFH, oEx.Message

Close #IFH

End Try

C. Dim oWriter As New \_ IO.StreamWriter(IO.File.Open(logfilePath, \_ IO.FileMode.Append)) Dim

oListener As New \_

TextWriterTraceListener(oWriter)

Debug.Listeners.Add(oListener)

Try

UpdateData()

Catch oEx As Exception

Debug.WriteLine(oEx.Message)

oWriter.Flush()

oWriter.Close()

oWriter = Nothing

End Try

D. Dim oWriter As New \_ IO.StreamWriter(IO.File.Open(logfilePath, \_ IO.FileMode.Append))

Dim oListener As New \_ TextWriterTraceListener(oWriter)

Trace.Listeners.Add(oListener)

Try

UpdateData()

Catch oEx As Exception

Trace.WriteLine(oEx.Message)

```
Trace.Listeners.Clear()
oWriter.Close() oWriter = Nothing
End Try
Answer: D
```

14. You use Visual Studio .NET to create a Windows-based data management application named MyApp. You implement the following code segment:

```
Dim oSwitch As New_
TraceSwitch("MySwitch",My Trace Switch")
Dim oWriter As New_
IO.Stream Writer(IO.File.Open("c:\MyApp.txt",_
IO.FileMode.Append))
Dim oListener As New_
TextWriterTraceListener(oWriter)
Trace.Listeners.Add(oListener)
Try
CustomerUpdate()
Catch oEx As Exception
Trace.WriteLineIf(oSwitch.TraceError,_
"Error:"& oEx.Message)
Finally
Trace.Listeners.Clear()
oWriter.Close()
oWriter = Nothing
End Try
```

You compile a debug version of the application and deploy it to a user's computer. The user reports errors, which are generated within the CustomerUpdate procedure.

You decide to enable logging of the error messages generated by CustomerUpdate. You want to use the minimum amount of administrative effort. What should you do?

- A. Start the application with the following command line: /TRACE MySwitch 1.
- B. Start the application with the following command line: /d:TRACE=True.
- C. Start the application with the following command line: /XML
- D. Create an environment variable on the user's computer. Name the variable MySwitch and assign it a value of 1.
- E. Edit your application's .config file to set the value of MySwitch to 1.

Answer: E

15. You develop a Windows-based application named CustOrders. You implement the Trace object within your application code. You will use this object to record application information, such as errors and performance data, in a log file. You must have the ability to enable and disable Trace logging. This

functionality must involve the minimum amount of administrative effort. What should you do?

- A. Create a Boolean constant in your application named #TraceLogging and set it to False . Each time your code uses Trace logging, use a #If?Then statement to evaluate your #TraceLogging constant.
- B. On each computer that will host your application, create an environment variable named CustOrders.Trace. Set the environment variable to True when you want to enable Trace logging. Set it to False when you want to disable Trace logging.
- C. On each computer that will host your application, edit the shortcut used to start your application. Add /d:TRACE=True to the Target property.
- D. Use the TraceSwitch class within your code. Each time your code uses Trace logging, consult the TraceSwitch level to verify whether to log information. Change the TraceSwitch level by editing your application's .config file.

Answer: D

16. You use Visual Studio .NET to create an application that tracks support incidents for your technical support department. You implement the Trace class to write information about run-time errors in a local log file. You also implement a TraceSwitch object named MySwitch, which can turn Trace logging on and off as needed. To maximize application performance, you ensure that MySwitch is disabled by default. You set your Configuration Manager to Release. You compile the application and deploy it to a shared folder on

your company intranet. Fifty users access the application from a shortcut on their desktops. One user receives error messages while running the application. You decide to enable verbose trace logging within the application for that user. You must ensure that you do not affect application performance for the other users. Which action or actions should you take? (Choose all that apply.)

- A. Set your Configuration Manager to Debug. Compile your application and deploy it locally on the user's computer. Create a new shortcut on the user's desktop to access the local copy of the application.
- B. Copy the deployed version of the application from the shared folder. Deploy it locally on the user's computer. Create a new desktop shortcut on the user's desktop to access the local copy of the application.
- C. Edit the .config file for the application on the user's computer to enable MySwitch with a value of 4.
- D. Edit the .config file for the application on the shared folder to enable MySwitch with a value of 4.
- E. Start the application with the /d:TRACE=TRUE command line option.
- F. Start the application with the /TRACE MySwitch 4 command line option.

Answer: BC

17. You use Visual Studio .NET to create an application that contains several Windows Forms. One of the Windows Forms uses the Trace class to write debugging information in a local log file. You need to compile your application so it can be released to users. You want to achieve this goal with the minimum amount of administrative effort. For performance reasons, you will disable the trace functionality before you deliver the application. You might need to enable the Tracing code after deployment. What should you do?

- A. Set your Configuration Manager to Release , and then compile your application.
- B. Use a TraceSwitch object that enables or disables your Trace code through the application's configuration file.

C. During deployment to users' computers, create a shortcut that starts your application with the /TRACE=False command line option.

D. During deployment to users' computers, create a shortcut that starts your application with the /d:TRACE=False command line option.

Answer: B

18. You develop an application that invokes a procedure named ProcessRecords. You implement the Trace class to log any errors thrown by ProcessRecords. You direct the Trace output to a local log file named ErrorLog.txt by using the following code segment:

```
Dim oWriter As New _&  
IO.StreamWriter(IO.File.Open(logfilepath,_ &,  
Trace.Listeners.Add(oListener)  
  
Try  
CustomerUpdate()  
Catch oExAs Exception  
Trace.WriteLineIf(oSwitch.TraceError,_  
"Error:"&oEx.Message)  
Finally  
Trace.Listeners.Clear()  
oWriter.Close()  
oWriter=Nothing  
End Try
```

Now you need to add code to your Finally construct to write all output in the ErrorLog.txt file and then close the file. You want to write the minimum amount of code to achieve this goal. Which code segment should you use?

- A. oWriter.Close()
- B. Trace.Flush() oWriter.Close()
- C. Trace.AutoFlush = True oWriter.Close()
- D. oWriter.AutoFlush = True oWriter.Close()

Answer: A

19. You develop a Windows-based application that processes customer data from a database. Your customer service representatives will use the application to view or edit customer data. One procedure in your application calls a function named ProcessCustomer. You want to add code to your application to record error information and performance data relating to ProcessCustomer. This information must be recorded in a log file named InfoLog.txt on the computer that executes the application. In addition, you

want to be able to enable and disable logging without recompiling your application, and you want the option of logging error information only, or of logging both error information and performance data. Which code segment should you use?

A. Dim oLog As New EventLog()

```
Dim oType As EventLogEntryType
```

```
Try
```

```
If Not EventLog.SourceExists("CustPerformance") Then
```

```
EventLog.CreateEventSource("CustPerformance", _ "Application")
```

```
End If
```

```
oLog.Source = "CustPerformance" oLog.Log = "Application"
```

```
oType = EventLogEntryType.Information
```

```
oLog.WriteEntry("Before ProcessCustomer() " _ & Now, oType)
```

```
ProcessCustomer()
```

```
If not EventLog.SourceExists("CustPerformance")
```

```
Then EventLog.CreateEventSource("CustPerformance", _ "Application")
```

```
End If
```

```
oLog.Source = "CustPerformance"
```

```
oLog.Log = "Application"
```

```
oType = EventLogEntryType.Information
```

```
oLog.WriteEntry("After ProcessCustomer() " _ & Now, oType)
```

```
Catch oEx As Exception
```

```
If Not EventLog.SourceExists("CustErrors")
```

```
Then EventLog.CreateEventSource("CustErrors", _ "Application")
```

```
End If
```

```
oLog.Source = "CustErrors"
```

```
oLog.Log = "Application"
```

```
oType = EventLogEntryType.Error
```

```
oLog.WriteEntry("Error details: " & _ oEx.Message, oType)
```

```
End Try
```

B. #Const MySwitch As Long

```
Dim IFH As Long IFH = FreeFile()
```

```
Open logFile For Append As #IFH
```

```
Try #If MySwitch = 4 Then Print #IFH,
```

```
"Before ProcessCustomer() " & Now
```

```
#End If ProcessCustomer()
```

```
#If MySwitch = 4 Then Print #IFH,
```

```
"After ProcessCustomer() " & Now #End
```

```
If Catch oEx As Exception
```

```
#If MySwitch = 1 Then Print #IFH, oEx.Message
```

```
#End If Close #IFH
```

```
End Try
```

C. Dim oWriter As New \_ IO.StreamWriter(IO.File.Open(logFile, \_ IO.FileMode.Append)) Dim oListener

```
As New TextWriterTraceListener(oWriter) Dim
```

```
oSwitch As New TraceSwitch("MySwitch", _ "MyTraceSwitch")Trace.Listeners.Add(oListener)
Try
Trace.WriteLineIf(oSwitch.TraceVerbose, _ "Before ProcessCustomer() " & Now) ProcessCustomer()
Trace.WriteLineIf(oSwitch.TraceVerbose, _ "After ProcessCustomer() " & Now)
Catch oEx As Exception
Trace.WriteLineIf(oSwitch.TraceError, _ oEx.Message)
Finally oWriter.Close()
oWriter = Nothing End Try
```

```
D. Dim oWriter As New _ IO.StreamWriter(IO.File.Open(logFile, _ IO.FileMode.Append)) Dim oLi stener
As New TextWriterTraceListener(oWriter)
Dim oSwitch As New TraceSwitch("MySwitch", _ "My TraceSwitch") Debug.Listeners.Add(oListener)
Try
Debug.WriteLineIf(oSwitch.TraceVerbose, _ "Before ProcessCustomer() " & Now) ProcessCustomer()
Debug.WriteLineIf(oSwitch.TraceVerbose, _ "After ProcessCustomer() " & Now)
Catch oEx As Exception
Debug.WriteLineIf(oSwitch.TraceError, _ oEx.Message)
Finally
oWriter.Close() oWriter = Nothing
End Try
Answer: C
```

20. You develop a Windows-based application that uses several functions to calculate a given inventory quantity. This quantity is stored in a variable named IQuantity. When you test your application, you discover that the value of IQuantity sometimes falls below zero. For debugging purposes, you want your application to generate an error message in such cases. You also want to be able to view the call stack to help identify the function call that is causing the miscalculation. You need to insert additional code after the calculation of IQuantity. Which code segment should you use?

- A. Trace.Assert(IQuantity >= 0, \_ "Inventory cannot be less than zero.")
- B. Trace.Assert(IQuantity < 0, \_ "Inventory cannot be less than zero.")
- C. Trace.Fail(IQuantity >= 0, \_ "Inventory cannot be less than zero.")
- D. Trace.WriteLineIf(IQuantity < 0, \_ "Inventory cannot be less than zero.")

Answer: A

21. Your project team uses Visual Studio .NET to create an accounting application. Each team member uses the Write method of both the Debug class and the Trace class to record information about application execution in the Windows 2000 event log. You are performing integration testing for the application. You need to ensure that only one entry is added to the event log each time a call is made to the Write method of either the Debug class or the Trace class. What are two possible code segments for you to use? (Each correct answer presents a complete solution. Choose two.)

- A. Dim myTraceListener As New \_ EventLogTraceListener("myEventLogSource")  
Trace.Listeners.Add(myTraceListener)
- B. Dim myDebugListener As New \_ EventLogTraceListener("myEventLogSource")

- Debug.Listeners.Add(myDebugListener)
- C. Dim myTraceListener As New \_ EventLogTraceListener("myEventLogSource")  
Debug.Listeners.Add(myTraceListener) Trace.Listeners.Add(myTraceListener)
- D. Dim myDebugListener As New \_ EventLogTraceListener("myEventLogSource") Dim  
myTraceListener As New \_  
EventLogTraceListener("myEventLogSource")  
Debug.Listeners.Add(myDebugListener) Trace.Listeners.Add(myTraceListener)

Answer: AB

22. You are maintaining a Visual Studio .NET sales application that was created by another developer. The application functions as expected for several months. Then users report that it sometimes calculates tax amounts incorrectly. An examination of the source code leads you to suspect that the errors are caused by a function named

CalculateSalesTax. To test your hypothesis, you place a breakpoint on the following line of code: `decTax = CalculateSalesTax(decRate, decSaleAmount)` However, when you run the application to create a test invoice, the breakpoint is not invoked. How should you correct this problem?

- A. Select Enable All Breakpoints from the Debug menu.
- B. Select Configuration Manager from the Build menu. Set the Active Solution Configuration option to Debug. Set the Configuration property of the project to Debug.
- C. Select Options from the Tools menu and then select the General object from the Debugging folder. Choose the option In break mode, only stop execution of the current process.
- D. Select Exceptions from the Debug menu. Under the heading If the exception is not handled, select Break into the Debugger.

Answer: B

23. You use Visual Studio .NET to create an accounting application, which includes a function named CreditCardValidate. This function contains several dozen variables and objects. To debug CreditCardValidate, you create a breakpoint at the top of the function. You run the accounting application within the Visual Studio .NET IDE and step through the code for CreditCardValidate. You need to examine the contents of the variables and objects in scope on each line of code. However, you want to avoid seeing the contents of all variables and objects within CreditCardValidate. You also need to complete the debugging process as quickly as possible. What should you do?

- A. Use the Autos window.
- B. Use the Locals window.
- C. Use the QuickWatch window.
- D. From the Command window, print the contents of each variable that you want to examine by using the following code segment: `? <variablename>`

Answer: A

24. You develop a Visual Studio .NET application that contains a function named CustomerUpdate. For debugging purposes, you need to add an entry to a log file whenever CustomerUpdate is executed. The log file is named DebugLog.txt. For maximum readability, you must ensure that each entry in DebugLog.txt appears on a separate line. Which code segment should you use?

- A. Dim oWriter As New IO.StreamWriter(IO.File.Open \_

- (logFile, IO.FileMode.Append)) Dim oListener As New \_ TextWriterTraceListener(oWriter)  
Debug.Listeners.Add(oListener) Debug.WriteLine("CustomerUpdate " & Now)
- B. Dim oWriter As New IO.StreamWriter(IO.File.Open \_  
(logFile, IO.FileMode.Append))  
Dim oListener As New \_  
TextWriterTraceListener(oWriter) Debug.Listeners.Add(oListener) Debug.Write("CustomerUpdate " &  
Now)
- C. Dim oListener As New TextWriterTraceListener()  
oListener.Name = logFile  
Debug.Listeners.Add(oListener)  
Debug.WriteLine("CustomerUpdate " & Now)
- D. Dim oListener As New TextWriterTraceListener()  
oListener.Name = logFile Debug.Listeners.Add(oListener) Debug.Write("CustomerUpdate " & Now)

Answer: A

25. You use Visual Studio .NET to create an application that interacts with a Microsoft SQL Server database. You create a SQL Server stored procedure named CustOrderDetails and save it in the database. Other developers on your team frequently debug other stored procedures. You need to verify that your stored procedure is performing correctly. You need to step through CustOrderDetails inside the Visual Studio .NET debugger. What should you do?

- A. Run CustOrderDetails by using the Visual Studio .NET Server Explorer.  
B. Step into CustOrderDetails by using the Visual Studio .NET Server Explorer.  
C. From the Command window, use Ctrl+E to run CustOrderDetails.  
D. Move CustOrderDetails from the Visual Studio.NET Server Explorer window to a Windows Form. Run the application in Debug mode and step through CustOrderDetails.

Answer: B

26. You use Visual Studio .NET to create a Windows Service application. You compile a debug version and install it on your computer, which runs Windows 2000 Server. You start the application from the Windows 2000 Service Control Manager. Now you need to begin debugging it within Visual Studio .NET. What should you do?

- A. Add a reference to the application within Visual Studio .NET. Add breakpoints to the code. Invoke the breakpoints by sending Stop, Pause, and Continue commands from the Service Control Manager.  
B. Select Processes from the Debug menu and attach the debugger to your application.  
C. Place a breakpoint in the OnStart method of the application and then run it.  
D. Place a breakpoint in the Main procedure of the application and then run it.

Answer: B

27. Your development team is using Visual Studio .NET to create an accounting application, which contains a substantial amount of code. Sometimes a developer cannot complete a portion of code until dependencies are completed by another developer. In these cases, the developer adds a comment in the application code about the work left incomplete. Such comments always begin with the text UNDONE. You open the application code on your computer. As quickly as possible, you need to view each comment that indicates incomplete work. What should you do?

- A. Configure the Task List window to show all tasks.
- B. Use the Build Comment Web Pages dialog box to build Comment Web Pages for the entire solution.
- C. Open a code window and use the Find in Files dialog box to search the application code for UNDONE.
- D. Open a code window and use the Find dialog box to search all open documents for UNDONE.

Answer: A

28. Your company assigns you to modify a Visual Studio .NET application that was created by a former colleague. However, when you try to build the application, you discover several syntax errors. You need to correct the syntax errors and compile a debug version of the code so the application can be tested. Before compiling, you want to locate each syntax error as quickly as possible. What should you do?

- A. Select each error listed in the Task List window.
- B. Open the Application event log from the Visual Studio .NET Server Explorer window. Select each error listed.
- C. Run the application in Debug mode. Each time an error is encountered, correct it and continue debugging the application.
- D. Select Build Solution from the Build menu. When the build fails, correct each error listed in the Output window.
- E. Select Build Comment Web Pages from the Tools menu. Select each function listed in the report that is generated.

Answer: A

29. Your development team uses Visual Studio .NET to create an accounting application, which contains a class named SupplierAccounts. This class instantiates several classes from a COM component that was created by using Visual Basic 6.0. Each COM component class includes a custom method named ShutDownObject that must be called before terminating references to the class. Software testers report that the COM component appears to remain in memory after the application terminates. You must ensure that the ShutDownObject method of each COM component class is called before SupplierAccounts is terminated. What should you do?

- A. Add code to the Terminate event of SupplierAccounts to call the ShutDownObject method of each COM component class.
- B. Find each location in your code where a reference to SupplierAccounts is set to Nothing or goes out of scope. Add code after each instance to manually invoke the Visual Studio .NET garbage collector.
- C. Add the procedure Protected Overrides Finalize() to SupplierAccounts. Add code to the procedure to call the ShutDownObject method of each COM component class.
- D. Add the procedure Private Sub Finally() to SupplierAccounts. Add code to the procedure to call the ShutDownObject method of each COM component class.

Answer: C

30. Your development team creates an order entry application by using Visual Studio .NET. The application stores and retrieves data in a Microsoft SQL Server database. All database connections in the application are centralized in class variables within a class named MyDataClass. Each time your application needs to access data from the database, it creates an instance of MyDataClass by using the following code segment:

Dim oData As New MyDataClass() When the oData variable is no longer needed, it is set to Nothing or goes out of scope. Initially, about 500 sales representatives use the application. Later, your company hires 50 new sales representatives who also use the application. You discover that the database is running out of available connections because of the increased usage. You must ensure that database connections are released immediately when they are no longer needed. You must also maintain an optimum level of application performance. What should you do?

A. Add the procedure

```
Protected Overrides Finalize()  
to MyDataClass.
```

Write code in the procedure to close all open database connections.

B. Add the procedure

```
Private Sub Finally()  
to MyDataClass.
```

Write code in the procedure to close all open database connections.

C. Implement the IDisposable interface within MyDataClass. Write code in the Dispose procedure of IDisposable to close all open database connections. Call the Dispose method of MyDataClass before any reference to MyDataClass is set to Nothing or goes out of scope.

D. Find each location in your code where a reference to MyDataClass is set to Nothing or goes out of scope. Add code after each instance to manually invoke the Visual Studio .NET garbage collector.

E. Add code to the Terminate event of MyDataClass to close all open database connections.

F. Ensure that each reference to MyDataClass is set to Nothing before it goes out of scope.

Answer: C

ExamSavior.com was founded in 2006. The safer,easier way to help you pass any IT Certification exams . We provide high quality IT Certification exams practice questions and answers(Q&A). Especially [Adobe](#), [Apple](#), [Citrix](#), [Comptia](#), [EMC](#), [HP](#), [HuaWei](#), [LPI](#), [Nortel](#), [Oracle](#), [SUN](#), [Vmware](#) and so on. And help you pass any IT Certification exams at the first try.

You can reach us at any of the email addresses listed below.

English Customer:

Sales : [sales@examsavior.com](mailto:sales@examsavior.com)

Support: [support@examsavior.com](mailto:support@examsavior.com)

Website: [www.examavior.com](http://www.examavior.com)

