

EXAMGOOD

QUESTION & ANSWER

Exam Good provides update free of charge in one year!

Accurate study guides
High passing rate!

<http://www.examgood.com>

Exam : **70-566**

Title : Upgrade: Transition your
MCPD Windows Developer
Skills to MCPD Windows
Developer 3

Version : Demo

1. You are creating a Windows Forms application by using the .NET Framework 3.5.

The application requires a form to display a clock.

You need to create a circular form to display the clock.

Which code segment should you use?

A. `this.FormBorderStyle =`

`System.Windows.Forms.FormBorderStyle.None;`

`System.Drawing.Drawing2D.GraphicsPath path = new`

`System.Drawing.Drawing2D.GraphicsPath();`

`path.AddEllipse(0, 0, this.Width, this.Height);`

`Region reg = new Region();`

`this.Region = reg;`

B. `this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedSingle;`

`System.Drawing.Drawing2D.GraphicsPath path = new`

`System.Drawing.Drawing2D.GraphicsPath();`

`path.AddEllipse(0, 0, this.Width, this.Height);`

`Region reg = new Region(path);`

`this.Region = reg;`

C. `this.FormBorderStyle =`

`System.Windows.Forms.FormBorderStyle.None;`

`System.Drawing.Drawing2D.GraphicsPath path = new`

`System.Drawing.Drawing2D.GraphicsPath();`

`path.AddEllipse(0, 0, this.Width, this.Height);`

`Region reg = new Region(path);`

`this.Region = reg;`

D. `this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedSingle;`

`System.Drawing.Drawing2D.GraphicsPath path = new`

`System.Drawing.Drawing2D.GraphicsPath();`

`path.AddEllipse(0, 0, this.Width, this.Height);`

`Region reg = new Region();`

`this.Region = reg;`

Answer: C

2.You are creating a Windows Forms application by using the .NET Framework 3.5.

The application requires a form to display a clock.

You need to create a circular form to display the clock.

Which code segment should you use?

A. `Me.FormBorderStyle = System.Windows.Forms.FormBorderStyle.None`

```
Dim path As New System.Drawing.Drawing2D.GraphicsPath()
```

```
path.AddEllipse(0, 0, Me.Width, Me.Height)
```

```
Dim reg As New Region()
```

```
Me.Region = reg
```

B. `Me.FormBorderStyle =`

```
System.Windows.Forms.FormBorderStyle.FixedSingle
```

```
Dim path As New System.Drawing.Drawing2D.GraphicsPath()
```

```
path.AddEllipse(0, 0, Me.Width, Me.Height)
```

```
Dim reg As New Region(path)
```

```
Me.Region = reg
```

C. `Me.FormBorderStyle = System.Windows.Forms.FormBorderStyle.None`

```
Dim path As New System.Drawing.Drawing2D.GraphicsPath()
```

```
path.AddEllipse(0, 0, Me.Width, Me.Height)
```

```
Dim reg As New Region(path)
```

```
Me.Region = reg
```

D. `Me.FormBorderStyle =`

```
System.Windows.Forms.FormBorderStyle.FixedSingle
```

```
Dim path As New System.Drawing.Drawing2D.GraphicsPath()
```

```
path.AddEllipse(0, 0, Me.Width, Me.Height)
```

```
Dim reg As New Region()
```

```
Me.Region = reg
```

Answer: C

3.You are creating a Windows Forms application by using the .NET Framework 3.5.

You create a new form in the application. You add a ContextMenuStrip control named ctxMenu to the form.

You have a user-defined class named CustomControl.

You write the following code segment in the application. (Line numbers are included for reference only.)

```
01 CustomControl myControl = new CustomControl();
```

```
02
```

You need to ensure that an instance of CustomControl is displayed on the form as a top-level item of the ctxMenu control.

Which code segment should you add at line 02

A. ToolStripControlHost host = new ToolStripControlHost(myControl);

```
ctxMenu.Items.Add(host);
```

B. ToolStripPanel panel = new ToolStripPanel();

```
panel.Controls.Add(myControl);
```

```
ctxMenu.Controls.Add(panel);
```

C. ToolStripContentPanel panel = new ToolStripContentPanel();

```
panel.Controls.Add(myControl);
```

```
ctxMenu.Controls.Add(panel);
```

D. ToolStripMenuItem menuItem = new ToolStripMenuItem();

```
ToolStripControlHost host = new ToolStripControlHost(myControl);
```

```
menuItem.DropDownItems.Add(host);
```

```
ctxMenu.Items.Add(menuItem);
```

Answer: A

4.You are creating a Windows Forms application by using the .NET Framework 3.5.

You create a new form in your application. You add a PrintDocument control named pntDoc to the form.

To support the print functionality, you write the following code segment in the application. (Line numbers are included for reference only.)

```
01 pntDoc.BeginPrint +=
```

```
    new PrintEventHandler(PrintDoc_BeginPrint);
```

```
02 ...
```

```
03 bool canPrint = CheckPrintAccessControl();
04 if (!canPrint) {
05
06 }
07
```

You need to ensure that the following requirements are met:

When the user has no print access, font and file stream initializations are not executed and the print operation is cancelled.

Print operations are logged whether or not the user has print access.

What should you do?

A. Add the following code segment at line 05.

```
pntDoc.BeginPrint -= new PrintEventHandler(PrintDoc_BeginPrint);
pntDoc.BeginPrint +=
    new PrintEventHandler((obj, args) => args.Cancel = true);
```

Add the following code segment at line 07.

```
pntDoc.BeginPrint +=
    new PrintEventHandler((obj1, args1) => LogPrintOperation());
```

B. Add the following code segment at line 05.

```
pntDoc.BeginPrint +=
    new PrintEventHandler(delegate(object obj, PrintEventArgs args){});
```

Add the following code segment at line 07.

```
pntDoc.BeginPrint -= new PrintEventHandler(PrintDoc_BeginPrint);
pntDoc.BeginPrint +=
    new PrintEventHandler((obj1, args1) => LogPrintOperation());
```

C. Add the following code segment at line 05.

```
pntDoc.BeginPrint -= new PrintEventHandler(PrintDoc_BeginPrint);
pntDoc.BeginPrint -=
    new PrintEventHandler(delegate(object obj, PrintEventArgs args){});
```

Add the following code segment at line 07.

```
pntDoc.BeginPrint -=
```

```
new PrintEventHandler((obj1, args1) => LogPrintOperation());
```

D. Add the following code segment at line 05.

```
pntDoc.BeginPrint -=
```

```
new PrintEventHandler((obj, args) => args.Cancel = true);
```

Add the following code segment at line 07.

```
pntDoc.BeginPrint += new PrintEventHandler(PrintDoc_BeginPrint);
```

```
pntDoc.BeginPrint -=
```

```
new PrintEventHandler((obj1, args1) => LogPrintOperation());
```

Answer: A

5. You are creating a Windows Forms application by using the .NET Framework 3.5.

You create a new form in your application. You add a PrintDocument control named pntDoc to the form.

To support the print functionality, you write the following code segment in the application. (Line numbers are included for reference only.)

```
01 AddHandler pntDoc.BeginPrint, _  
    AddressOf PrintDoc_BeginPrint
```

```
02 ...
```

```
03 Dim canPrint As Boolean = CheckPrintAccessControl()
```

```
04 If canPrint = False Then
```

```
05
```

```
06 End If
```

```
07
```

You need to ensure that the following requirements are met:

When the user has no print access, font and file stream initializations are not executed and the print operation is cancelled.

Print operations are logged whether or not the user has print access.

What should you do?

A. Add the following code segment at line 05.

```
RemoveHandler pntDoc.BeginPrint, AddressOf PrintDoc_BeginPrint
```

```
AddHandler pntDoc.BeginPrint, _
```

```
Function(obj1, args1) args1.Cancel = True
```

Add the following code segment at line 07.

```
AddHandler pntDoc.BeginPrint, AddressOf
```

```
LogPrintOperation
```

B. Add the following code segment at line 05.

```
AddHandler pntDoc.BeginPrint, AddressOf EmptyEventHandler
```

Add the following code segment at line 07.

```
RemoveHandler pntDoc.BeginPrint, AddressOf PrintDoc_BeginPrint
```

```
AddHandler pntDoc.BeginPrint, AddressOf
```

```
LogPrintOperation
```

C. Add the following code segment at line 05.

```
RemoveHandler pntDoc.BeginPrint, AddressOf PrintDoc_BeginPrint
```

```
RemoveHandler pntDoc.BeginPrint, AddressOf EmptyEventHandler
```

Add the following code segment at line 07.

```
RemoveHandler pntDoc.BeginPrint, AddressOf
```

```
LogPrintOperation
```

D. Add the following code segment at line 05.

```
AddHandler pntDoc.BeginPrint, _
```

```
Function(obj1, args1) args1.Cancel = True
```

Add the following code segment at line 07.

```
AddHandler pntDoc.BeginPrint, AddressOf PrintDoc_BeginPrint
```

```
RemoveHandler pntDoc.BeginPrint, AddressOf
```

```
LogPrintOperation
```

Answer: A

6. You are creating a Windows Forms application by using the .NET Framework 3.5.

You plan to modify a list of orders within a DataGridView control in the application.

You need to ensure that a value is required in the first column of the grid control.

Which code segment should you use?

A. private void dataGridOrders_CellValidated(


```
object sender, DataGridViewCellEventArgs e) {  
    if (e.ColumnIndex == 0) ? {  
        var cellValue = dataGridOrders[  
            e.ColumnIndex, e.RowIndex].Value;  
        if (cellValue == null ||  
            string.IsNullOrEmpty(cellValue.ToString()))  
            {  
                dataGridOrders.EndEdit();  
            }  
    }  
}
```

```
B. private void dataGridOrders_Validated(  
    object sender, EventArgs e) {  
        if (dataGridOrders.CurrentCell.ColumnIndex == 0) {  
            var cellValue = dataGridOrders.Text;  
            ?if (cellValue == null ||  
                ?string.IsNullOrEmpty(cellValue.ToString()))  
                {  
                    ?dataGridOrders.EndEdit();  
                }  
        }  
    }  
}
```

```
C. private void dataGridOrders_Validating(  
    object sender, CancelEventArgs e) {  
        if (dataGridOrders.CurrentCell.ColumnIndex == 0) {  
            var cellValue = dataGridOrders.Text;  
            ?if (cellValue == null ||  
                string.IsNullOrEmpty(cellValue.ToString()))  
                ?{  
                    e.Cancel = true;  
                }  
        }  
    }  
}
```

```
        ?}  
    }  
}  
D. private void dataGridViewOrders_CellValidating(  
    object sender, DataGridViewCellValidatingEventArgs e) {  
    if (e.ColumnIndex == 0) {  
        if (e.FormattedValue == null ||  
            string.IsNullOrEmpty(e.FormattedValue.ToString()))  
            ?{  
                e.Cancel = true;  
            ?}  
    }  
}
```

Answer: D

7.You create a Windows application by using the .NET Framework 3.5. The application is named APP1.

APP1 has the following specifications:

It merges information from text files.

It reads each text file as a string.

It runs on a server computer named Server1.

Another application named APP2 also runs on Server1.

You discover that when a function is used to merge the text files, APP1 consumes additional memory.

This additional memory consumption affects the operations of APP2.

You need to modify the design of APP1 to ensure minimum effect on APP2.

What should you do?

A. Use the + operator for the merge process.

B. Use the StringBuilder.Append method for the merge process.

C. Write a code segment to perform the merge process on a new thread.

D. Write a code segment in a separate project to perform the merge process. Configure the project to start the merge operation as a new process.

Answer: B

8.You are creating a Windows Forms application by using the .NET Framework 3.5.

You plan to modify a list of orders within a DataGridView control in the application.

You need to ensure that a value is required in the first column of the grid control.

Which code segment should you use?

```
A. Private Sub dataGridOrders_CellValidated( _  
    ByVal sender As Object, _  
    ByVal e As DataGridViewCellEventArgs) _  
    Handles dataGridOrders.CellValidated  
    If e.ColumnIndex = 0 Then  
        Dim cellValue = dataGridOrders(e.ColumnIndex, e.RowIndex).Value  
        If cellValue = Nothing _  
        Or String.IsNullOrEmpty(cellValue.ToString()) Then  
            dataGridOrders.EndEdit()  
        End If  
    End If  
End Sub
```

```
B. Private Sub dataGridOrders_Validated( _  
    ByVal sender As Object, _  
    ByVal e As EventArgs) _  
    Handles dataGridOrders.Validated  
    If dataGridOrders.CurrentCell.ColumnIndex = 0 Then  
        Dim cellValue = dataGridOrders.Text  
        If cellValue = Nothing Or _  
        String.IsNullOrEmpty(cellValue.ToString()) Then  
            dataGridOrders.EndEdit()  
        End If  
    End If  
End Sub
```

```
C. Private Sub dataGridOrders_Validating( _  
    ByVal sender As Object, _  
    ByVal e As CancelEventArgs) _  
    Handles dataGridOrders.Validating  
    If dataGridOrders.CurrentCell.ColumnIndex = 0 Then  
        Dim cellValue = dataGridOrders.Text  
        If cellValue = Nothing Or _  
        String.IsNullOrEmpty(cellValue.ToString()) Then  
            e.Cancel = True  
        End If  
    End If  
End Sub
```

```
D. Private Sub dataGridOrders_CellValidating( _  
    ByVal sender As Object, _  
    ByVal e As DataGridViewCellValidatingEventArgs) _  
    Handles dataGridOrders.CellValidating  
    If e.ColumnIndex = 0 Then  
        If e.FormattedValue = Nothing _  
        Or String.IsNullOrEmpty(e.FormattedValue.ToString()) Then  
            e.Cancel = True  
        End If  
    End If  
End Sub
```

Answer: D

9. You are creating a Windows Forms application by using the .NET Framework 3.5.

You write the following code segment to bind a list of categories to a drop-down list. (Line numbers are included for reference only.)

```
01 OleDbConnection cnnNorthwind =  
    new OleDbConnection(connectionString);
```

```
02 OleDbCommand cmdCategory = new OleDbCommand(
    "SELECT CategoryID, CategoryName FROM Categories ORDER BY
    CategoryName", cnnNorthwind);
03 OleDbDataAdapter daCategory = new
    OleDbDataAdapter(cmdCategory);
04 DataSet dsCategory = new DataSet();
05 daCategory.Fill(dsCategory);
06
```

You need to ensure that the drop-down list meets the following requirements:

Displays all category names.

Uses the category ID as the selected item value.

Which code segment should you add at line 06?

- A. `ddlCategory.DataSource = dsCategory;`
`ddlCategory.DisplayMember = "CategoryName";`
`ddlCategory.ValueMember = "CategoryID";`
- B. `ddlCategory.DataSource = dsCategory.Tables[0];`
`ddlCategory.DisplayMember = "CategoryName";`
`ddlCategory.ValueMember = "CategoryID";`
- C. `ddlCategory.DataBindings.Add("DisplayMember",`
`dsCategory, "CategoryName");`
`ddlCategory.DataBindings.Add("ValueMember",`
`dsCategory, "CategoryID");`
- D. `ddlCategory.DataBindings.Add("DisplayMember",`
`dsCategory.Tables[0], "CategoryName");`
`ddlCategory.DataBindings.Add("ValueMember",`
`dsCategory.Tables[0], "CategoryID");`

Answer: B

10. You are creating a Windows Forms application by using the .NET Framework 3.5.

You write the following code segment to update multiple databases on a SQL Server 2008 database.

(Line numbers are included for reference only.)

```
01 string connectionStringCustomer = @"Data
    Source=CUSTOMER;Integrated Security= SSPI;";
02 string connectionStringOrders = @"Data Source=ORDER
    ;Integrated Security= SSPI;";
03 SqlCommand cmdCustomer = new SqlCommand();
04 SqlCommand cmdOrders = new SqlCommand();
05 SqlConnection cnnCustomer =
    new SqlConnection(connectionStringCustomer);
06 SqlConnection cnnOrders =
    new SqlConnection(connectionStringOrders);
07
```

You need to ensure that all database updates are included in a single distributed transaction.

Which code fragment should you add on Line 07?

A. `cnnCustomer.Open();`

`cnnOrders.Open();`

...

`cmdOrders.ExecuteNonQuery();`

...

`cmdCustomer.ExecuteNonQuery();`

`cnnOrders.Close();`

`cnnCustomer.Close();`

B. `TransactionScope scope = new TransactionScope();`

`cnnCustomer.Open();`

`cnnOrders.Open();`

...

`cmdOrders.ExecuteNonQuery();`

...

`cmdCustomer.ExecuteNonQuery();`

`cnnOrders.Close();`

```
cnnCustomer.Close();
```

```
scope.Complete();
```

```
C. TransactionScope customerScope =
```

```
new TransactionScope() {
```

```
using (SqlConnection cnnCustomer =
```

```
?new SqlConnection (connectionStringCustomer)) { }
```

```
customerScope.Complete(); }
```

```
using (TransactionScope ordersScope =
```

```
new TransactionScope()) {
```

```
using (SqlConnection cnnOrders =
```

```
new SqlConnection(connectionStringOrders)) { }
```

```
ordersScope.Complete(); }
```

```
D. try {
```

```
cmdOrders.Transaction = cnnOrders.BeginTransaction();
```

```
...
```

```
cmdOrders.ExecuteNonQuery();
```

```
...
```

```
cmdCustomer.Transaction = cnnCustomer.BeginTransaction();
```

```
...
```

```
cmdCustomer.ExecuteNonQuery();
```

```
cmdCustomer.Transaction.Commit();
```

```
cmdOrders.Transaction.Commit();
```

```
}catch {
```

```
cmdCustomer.Transaction.Rollback();
```

```
cmdOrders.Transaction.Rollback();
```

```
}
```

Answer: B