

Exam : Microsoft 70-547(C++)

**Title : PRO:Design and Develop
Web-Basd Apps by Using
MS.NET Frmwk**

Version : Demo



Top 3 Customers in Test4pass - Best IT certifications Team



Abbigail, who is proficient in Microsoft technology. After read our Study guides of Microsoft , She think test4pass is the best for IT candidates.

is the best for IT candidates.



Ramsden, who was majored in Manager IT technologys. He has many years of education experience. With his help, many of his candidates have won the certificate. he said:

test4pass' materials are the real one!



The IT experts of American company are training with test4pass braindumps, the manager of the company said: that will help them a lot

by using test4pass.

HOT Certifications On Test4pass

Cisco Certifications

[CCNA](#) [CCDA](#) [CCNP](#) [CCDP](#) [CCVP](#) [CCSP](#) [CCIP](#) [CCIE](#) [CCDE](#) [Data Center](#) [Sales Expert](#)

Microsoft Certifications

[MCP](#) [MCSE](#) [MCITP](#) [MCTS](#) [MCSE2003](#) [MCPD](#) [MCSA.NET](#) [MCDST](#) [TS](#) [Exchange Server2007](#)
[MCSE2003 Security](#) [MCSE2003 Messaging](#) [Microsoft Business Solutions](#)

IBM Certifications

[Certified Administrator](#) [Certified Systems Expert](#) [Solutions Expert](#) [System Administrator](#)
[DB2](#) [Certified Advanced Technical Expert](#) [Certified Advanced System Administrator](#)
[Lotus Certification](#) [WebSphere](#) [Cognos 8 BI](#) [Certified Associate Developer](#) [Tivoli Software](#)

CompTIA Certifications

[A+](#) [CDIA+](#) [CTT+](#) [e-Biz+](#) [CompTIA HTI+](#) [i-NET+](#) [Linux+](#) [Network+](#) [Project+](#) [RFID+](#) [Security+](#)
[Server+](#) [PDI+](#) [Convergence+](#)

1. You create Web-based applications. You are creating an Internet banking application. The application will be used by bank account holders.

You are creating a method to withdraw money from an account. The method must change the account balance according to one of the following rules:

If the amount that is being withdrawn is less than or equal to the account balance, then subtract the amount from the balance.

If the amount that is being withdrawn is greater than the account balance by up to 500 dollars, then subtract the amount and a 35-dollar fee from the balance.

If the amount that is being withdrawn is greater than the account balance by more than 500 dollars, then generate an error.

You are translating the specification given here into pseudo code. You start by writing the following code.

Method

```
public void Withdraw
```

Input parameters

```
decimal amount
```

Class field

```
decimal balance
```

Pseudo code

```
//your pseudo code
```

You need to insert the correct pseudo code.

Which code segment should you insert?

A. If amount < balance then balance - = amount

 If amount < balance + 500 then balance = balance - (amount + 35)

 If amount > balance + 500 then throw exception

B. If amount <= balance then balance - = amount

 If amount <= balance + 500 then balance = balance - (amount + 35)

 If amount > balance + 500 then throw exception

C. If amount < balance then balance - = amount

 Else If amount < balance + 500 then balance = balance - (amount + 35)

 Else throw exception

D. If amount <= balance then balance - = amount

 Else If amount <= balance + 500 then balance = balance - (amount + 35)

 Else throw exception

Answer: D

2. You create Web-based client applications. You create a component named Orders for a company named

Northwind Traders.

This component is used to retrieve and update data in the Orders table of the company's database. The schema of the Orders table is as shown in the following Exhibit. (Click the Exhibit button.)

OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate	ShippedDate	ShipVia	Freight	ShipName	ShipAddress	ShipCity	ShipRegion	ShipPostalCode	ShipCountry
PK													

The Orders component permits the client application to perform the following tasks:

Retrieve all orders for a Customer.

Retrieve all orders for an Employee.

Retrieve all orders that are shipped.

An instance of the Order class represents a single order that is identified by the OrderID parameter. An instance of the Order class permits the client application to perform the following tasks:

Ascertain whether an order is shipped.

Modify the RequiredDate field of the Order table for an existing order.

Delete an order.

You need to create the design for the component.

What should you do?

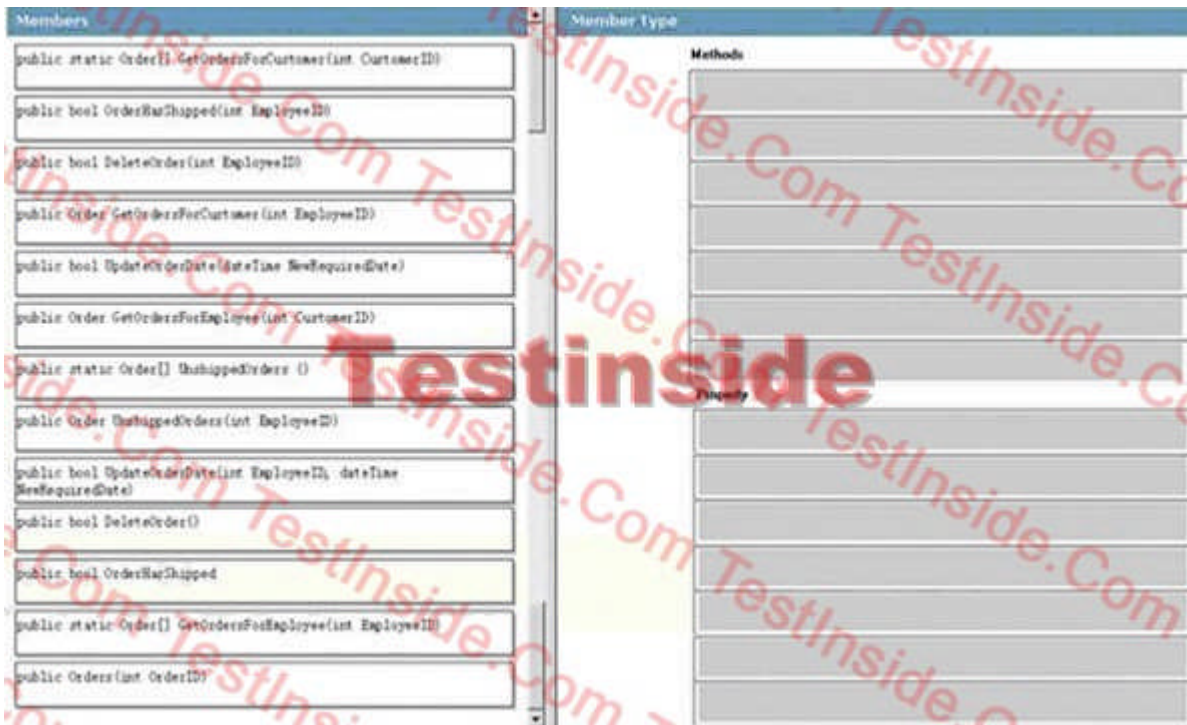
To answer, drag the appropriate members to the correct locations in the member type column.

Members

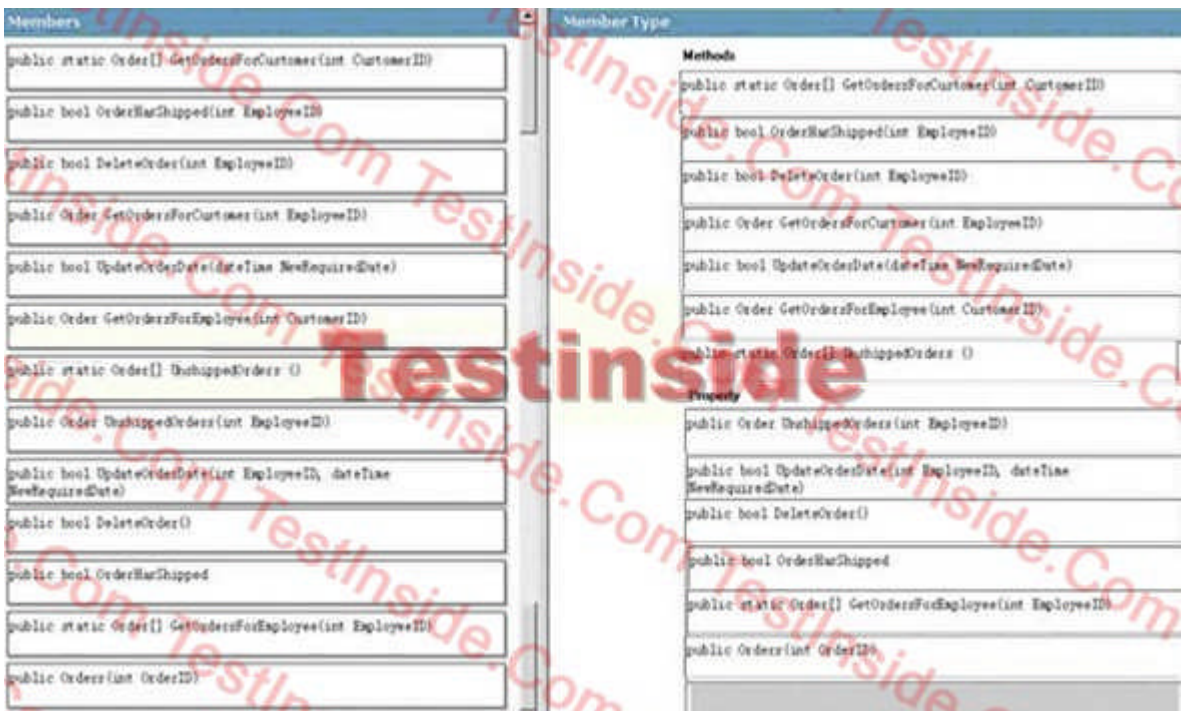
```

public static Order[] GetOrdersForCustomer(int CustomerID)
public bool OrderHasShipped(int EmployeeID)
public bool DeleteOrder(int EmployeeID)
public Order GetOrdersForCustomer(int EmployeeID)
public bool UpdateOrderDate(dateTime NewRequiredDate)
public Order GetOrdersForEmployee(int CustomerID)
public static Order[] UnshippedOrders ()
public Order UnshippedOrders(int EmployeeID)
public bool UpdateOrderDate(int EmployeeID, dateTime NewRequiredDate)
    
```

```
public bool DeleteOrder()
public bool OrderHasShipped
public static Order[] GetOrdersForEmployee(int EmployeeID)
public Orders(int OrderID)
```



Answer:



3. You create Web-based client applications. You create a class library that is named Fabrikam.dll. Ten applications will use Fabrikam.dll.

Fabrikam.dll contains two classes that are named Order and OrderDetail. The class library must meet the

following requirements:

Both the classes in Fabrikam.dll are available to client applications of Fabrikam.dll.

Each instance of the OrderDetail class is associated with an instance of the Order class.

Code segments in client applications do not instantiate the OrderDetail class.

The OrderDetail class contains no static members.

You need to design the interface for the OrderDetail class.

Which code segment should you choose?

A. public sealed class OrderDetail?{

```
    internal OrderDetail(){
```

```
        ...
```

```
    }
```

```
}
```

B. internal sealed class OrderDetail?{

```
    internal OrderDetail() {
```

```
        ...
```

```
    }
```

```
}
```

C. public sealed class OrderDetail?{

```
    public OrderDetail() {
```

```
        ...
```

```
    }
```

```
}
```

D. public sealed class OrderDetail?{

```
    private OrderDetail() {
```

```
        ...
```

```
    }
```

```
}
```

Answer: A

4. You create Web-based client applications. You are creating a class named Product. The Product class will be used by a Web-based application to retrieve and modify product information.

When you create an instance of the Product class, you retrieve the current information from the Products table. The Product class contains a static member named CreateNewProduct. The CreateNewProduct method is used to add a new product to the database and return the primary key. The Products table contains the following fields:

ProductID (primary key)

ProductName

Description

CategoryID

CurrentPrice

You need to create the constructor for the Product class.

Which code segment should you use?

A. `public Product(int ProductID, string ProductName, string Description, int CategoryID, decimal CurrentPrice){`

`...`

`}`

B. `public Product(int ProductID, string ProductName)?{`

`...`

`}`

C. `public Product()?{`

`...`

`}`

D. `public Product(int ProductID)?{`

`...`

`}`

Answer: D

5. You create components for Web-based client applications. You are creating a BankAccount class.

The BankAccount class contains an AccountNumber property and a CreateAccount method. The CreateAccount method is used to create a new account. The method generates a unique random value for the actNumber field.

You need to ensure that the BankAccount class is extendable, and that it serves as the base class for other derived classes. You also need to ensure that each derived class can have its own guidelines to generate account numbers in the CreateAccount method.

Which code segment should you use?

A. `public class BankAccount {
 protected long actNumber;
 public long AccountNumber {
 get {
 return actNumber;
 }
 }
}`

```
public virtual BankAccount CreateAccount() {  
    ...  
}  
}
```

```
B. public class BankAccount {  
    private long actNumber;  
    public long AccountNumber {  
        get {  
            return actNumber;  
        }  
    }  
    public virtual BankAccount CreateAccount() {  
        ...  
    }  
}
```

```
C. public class BankAccount {  
    protected long actNumber;  
    public long AccountNumber {  
        get {  
            return actNumber;  
        }  
    }  
    public BankAccount CreateAccount() {  
        ...  
    }  
}
```

```
D. public class BankAccount {  
    private long actNumber;  
    public long AccountNumber {  
        get {  
            return actNumber;  
        }  
    }  
    public BankAccount CreateAccount() {  
        ...  
    }  
}
```

```

    }
}

```

Answer: A

6. You create Web-based client applications. You are creating a class library that will be used by an e-commerce Web-based application. The library has an abstract class that is named Product. The Product class serves as a base class for the other classes and provides a default ProductID property.

Each class other than the base class represents a type of product that is sold by your company. There is a ProductID property and a GetProductDetails procedure for each product type.

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

The shopping cart in your Web-based application processes all product types in the same manner.

Each class retrieves its data from a different source.

The GetProductDetails procedure retrieves the data from the appropriate source for the product type.

What should you include in the Product class?

- A. an abstract ProductID property and a virtual GetProductDetails procedure
- B. a virtual ProductID property and a virtual GetProductDetails procedure
- C. a virtual ProductID property and an abstract GetProductDetails procedure
- D. an abstract ProductID property and an abstract GetProductDetails procedure

Answer: C

7. You create Web-based applications. You create a loan application form.

The loan application form is used to calculate the monthly payment of loans. The monthly payment is based on the loan amount, rate, and number of months. The form contains four text boxes and a button. There are no other controls in the form. The application event handler has the following lines of code. (Line numbers are included for reference only.)

```

01    protected void GetPayment(object sender, EventArgs e) {
02        try {
03            decimal rate = decimal.Parse(txtRate.Text);
04            decimal loanAmount = decimal.Parse(txtLoan.Text);
05            int period = int.Parse(txtPeriod.Text);
06            decimal payment = CalcPayment(rate, loanAmount, period);
07            txtPayment.Text = payment.ToString("C");
08        }
09        catch (OverflowException ex) {
10            ...
11        }
12        catch (InvalidCastException ex) {

```

```

13     ...
14     }
15     catch (Exception ex) {
16     ...
17     }
18     }

```

You must prevent exceptions whenever possible to meet the application requirements.

You need to evaluate the current exception handling mechanism.

What should you conclude?

- A. The current exception handling mechanism meets the requirements. Nothing needs to be changed.
- B. The current exception handling mechanism does not meet the requirements. A required field validator and a range validator control must be added to validate each text box before the button is clicked.
- C. The current exception handling mechanism does not meet the requirements. A required field validator control must be added to validate each text box before the button is clicked.
- D. The current exception handling mechanism does not meet the requirements. A regular expression validator control must be added to validate each text box before the button is clicked.

Answer: B

8. You create Web-based client applications. You are reviewing a Web application page that populates the list of all employees for your company.

The following code segment loads the list of employees from a database.

```

private static List<CEmployee> LoadEmployeesFromDatabase() {?
    DbProviderFactory factory =?
        DbProviderFactories.GetFactory("System.Data.SqlClient");
    List<CEmployee> lstEmployees = null;
    // Extract the connection string from configuration data
    ConnectionStringSettings connString =
        ConfigurationManager.ConnectionStrings["EmployeeStore"];
    // Create the connection and open it
    DbConnection conn = factory.CreateConnection();
    conn.ConnectionString = connString.ConnectionString;
    conn.Open();
    // Get the employees. The connection to the database is?
    // given as parameter
    lstEmployees = GetEmployees(conn);
    // Close the connection to the employee data store

```

```
Conn.Close();  
return lstEmployees;  
}
```

You analyze the code segment. You find that the database connection fails to close properly when the GetEmployees method throws an exception.

You need to recommend a change in the code segment to ensure that every possible code path closes the database connection.

Which code segment should you recommend?

A. // Create the connection and open it

```
using (DbConnection conn = factory.CreateConnection())  
{  
    conn.ConnectionString = connString.ConnectionString;  
    conn.Open();  
    // Get the employees. The connection to the database
```

// is given as parameter

```
    lstEmployees = GetEmployees(conn);  
}
```

B. // Create the connection and open it

```
DbConnection conn = factory.CreateConnection();  
conn.ConnectionString = connString.ConnectionString;  
conn.Open();  
// Get the employees. The connection to the database is
```

// given as parameter

```
lstEmployees = GetEmployees(conn);  
if (lstEmployees == null) {  
    conn.Dispose();  
}  
else {  
    conn.Close();  
}
```

C. HandleCollector coll =new HandleCollector("Connections", 0, 5);

// Create the connection and open it

```
DbConnection conn = factory.CreateConnection();  
conn.ConnectionString = connString.ConnectionString;  
conn.Open();
```

```
coll.Add();
// Get the employees. The connection to the database is
// given as parameter
lstEmployees = GetEmployees(conn);
// Close the connection to the employee data store
Conn.Close();
Coll.Remove();
D. // Create the connection and open it
using (IDisposable factory = DbProviderFactories.GetFactory("System.Data.SqlClient") as
IDisposable)
{
    DbConnection conn = factory.CreateConnection();
    conn.ConnectionString = connString.ConnectionString;
    conn.Open();
    // Get the employees. The connection to the database
// is given as parameter
    lstEmployees = GetEmployees(conn);
}
```

Answer: A

9. You create Web-based client applications. You are reviewing a Web application page that populates a list of all employees of your company.

You analyze code and find that the Web application page does not prevent exceptions from traveling to the browser.

You need to ensure that the Web application page intercepts exceptions and presents an error message to the browser.

What change should you suggest?

A. Add the following code segment to the Web.config file.

```
<system.web>
    <compilation debug="true"/>
</system.web>
```

B. Add the following code segment to the page.

```
protected void Page_Error(object sender, EventArgs e) {
    Response.Redirect("error.aspx");
}
```

C. Add the following code segment to the Web.config file.

```
<system.web>
    <customErrors mode="Off"/>
</system.web>
```

D. Change the load event handler to the following code segment.

```
protected void Page_Load(object sender, EventArgs e) {
    try {
        LoadEmployees();
    }
    Catch {
        Response.Redirect("error.aspx");
    }
}
```

Answer: B

10. You create Web-based client applications. You create a Web site that will be used to simulate different types of loans. You are writing a method to calculate the payment on a simple loan.

You write the following lines of code for the method. (Comments are included for reference only.)

```
public static decimal Payment(decimal loanAmount, int period, decimal rate) {
    if (!(loanAmount > 0)||!(period > 1)||!(rate > 0)) { // Line A
        throw new Exception("Invalid input!"); // Line B
    } else {
        return 0M; // Line C: return a calculated payment
    }
}

public static decimal CheckBalance(ulong accountID) {
    return 0M; // Line D: return calculated balance
}
```

You write the following code for the unit test.

```
[TestMethod()]
public void PaymentTest() {
    decimal payment = Loan.Payment(100000,360,10); // Line E
    Assert.AreEqual(payment, 877.57M); // Line F
}
```

You enable coverage testing for this unit test.

You need to identify the coverage of your test.

Which lines are covered by the test?

- A. Lines commented A, B, and C
- B. Lines commented A and C
- C. Lines commented A, B, C, D, E, and F
- D. Lines commented A, B, C, E, and F

Answer: B

11. You create Web-based applications. You are creating an Internet banking application.

You write the following lines of code to represent a method in your application. (Line numbers are included for reference only.)

```
01     public void Transfer(decimal amount, BankAccount account)
02     {
03         if (!(amount > 0))
04             throw new Exception("Invalid deposit amount!");
05         else
06         {
07             this.Withdraw(amount);
08             account.Deposit(amount);
09         }
10     }
```

You use the Microsoft Visual Studio 2005 test feature to automatically generate the following unit test. (Line numbers are included for reference only.)

```
01     [TestMethod()]
02     public void TransferTest()
03     {
04         BankAccount target = new BankAccount();
05         BankAccount transferTo = new BankAccount();
06         target.Deposit(500);
07         target.Transfer(100,transferTo);
08         Assert.Inconclusive("A method that does not return a value cannot be verified.");
09     }
10 }
```

You need to change the test method to return a conclusive result.

Which line of code should replace the code on line 09 of the unit test?

- A. Assert.AreEqual (400M, target.Balance);
- B. Assert.IsTrue (target.Balance != 400M);
- C. Debug.Assert (target.Balance == 400M,passed);
- D. Debug.Assert (target.Balance == 400M,failed);

Answer: A

12. You create Web-based client applications. You are creating an application that must access different databases depending on whether the application is in demonstration, production, or test mode. The mode is stored in an environment variable.

?

The configuration file contains the following settings.

?

```
<connectionStrings>
```

```
  <add name = "ProdDB" connectionString = "Data Source = server; Initial Catalog = ProdDatabase; Integrated_Security = true" providerName = "System.Data.SqlClient" />
```

```
  <add name = "DemoDB" connectionString = "Data Source = server; Initial Catalog = DemoDatabase; Integrated_Security = true" providerName = "System.Data.SqlClient" />
```

```
  <add name = "TestDB" connectionString = "Data Source = server; Initial Catalog = TestDatabase; Integrated_Security = true" providerName = "System.Data.SqlClient" />
```

```
</connectionStrings>
```

?

You need to ensure that changing the mode will not require you to change the configuration file.

?

Which code segment should you use?

A. public string GetConnectionString() {

```
#if PROD
```

```
    ?return WebConfigurationManager.ConnectionStrings["ProdDB"].
    ConnectionString;
```

```
#elseif DEMO
```

```
    ?return WebConfigurationManager.ConnectionStrings["DemoDB"].
    ConnectionString;
```

```
#else
```

```
    ?return WebConfigurationManager.ConnectionStrings["TestDB"].
    ConnectionString;
```

```
#endif
```

```
}
```

B. public string GetConnectionString() {

```
    return WebConfigurationManager.ConnectionStrings[
    Environment.GetEnvironmentVariable("WebAppModeDB")].
```

```
ConnectionString;
}
C. public string GetConnectionString() {
    return ConfigurationSettings.AppSettings[
        Environment.GetEnvironmentVariable("WebAppModeDB")];
}
D. public string GetConnectionString() {
#if PROD
?    return ConfigurationSettings.AppSettings["ProdDB"];
#elseif DEMO
    return ConfigurationSettings.AppSettings["DemoDB"];
#else
    return ConfigurationSettings.AppSettings["TestDB"];
#endif
}
E. public string GetConnectionString() {
    return WebConfigurationManager.ConnectionStrings["Database"].
        ConnectionString;
}
```

Answer: B

13. Question

You create Web-based client applications.

?

You are evaluating the design of an e-commerce Web site. The Web site processes credit card information. The Web site has a shopping cart and expects a high volume of traffic, especially during peak shopping times.

?

The design specifications for the application must meet the following criteria:

?

The application will be hosted on a Web farm.

The application will use SSL during the checkout process.

Shopping cart information will be stored in InProc session variables.

?

You need to evaluate the design of the application and recommend whether it is technically feasible and complete.

?

What should you conclude?

A. The design is technically feasible, but it is not complete. The application must be configured to use cookieless sessions. Each server on the farm must use a unique certificate.

B. The design is technically feasible and complete.

C. The design is technically feasible, but it is not complete. The servers must have their affinity set to a single host (sticky sessions).

D. The design is not technically feasible. The application cannot be hosted on a Web farm.

Answer: C

14. Question

You create Web-based client applications. You are designing a new Web site for your company. You need to evaluate the design concept for the main page, default.aspx.

?

The default.aspx page contains static information, except for one data grid that displays information about the companys products. The data grid will retrieve the data from a component. The component will retrieve the data from a stored procedure. The page has the appropriate permissions to the Microsoft SQL Server database.

?

The application design must achieve the following goals without altering the data component:

?

The default.aspx page must load quickly.

The default.aspx page must contain the latest data from the database.

?

You decide to use a sqlCacheDependency attribute to ensure that the default.aspx page always displays the latest data in the grid.

?

You need to ascertain whether the use of the sqlCacheDependency attribute meets the requirements of this application.

?

Which rationale should you choose?

A. The use of the sqlCacheDependency attribute is not feasible. The sqlCacheDependency attribute must be used in the component and not on the page.

B. The use of the sqlCacheDependency attribute is not feasible. The sqlCacheDependency attribute cannot

be used with a stored procedure.

C. The use of the sqlCacheDependency attribute is feasible. The sqlCacheDependency attribute can be used by including a sqlDependency attribute in the @OutputCache directive of the page.

D. The use of the sqlCacheDependency attribute is feasible. The sqlCacheDependency attribute can be linked to the data component that is used on the page.

Answer: C

15. Question

You create Web-based client applications. You are designing an extranet site for a company of trading partners.

?

You decide to use the following technologies:

?

Windows Authentication

XML to transfer data between the company and the traders

?

Before implementation, these technologies must be validated. You propose the following approach to perform the validation:

?

Coordinate a test set of Active Directory accounts for one trading partner.

Provide the trading partner access to a test site that has logon facility.

Distribute the XML schema that permits the partner to access data.

?

You need to evaluate whether the approach validates the proposed technology successfully.

?

What should you conclude?

A. The approach validates the use of the proposed technology for the application.

B. The approach does not validate the use of the proposed technology for the application. The company and the partner need to create test applications. The test applications need to read data in the proposed XML schema to establish the validation.

C. The approach does not validate the use of the proposed technology for the application. The company and the partner need to coordinate their Active Directory tree into a shared Active Directory forest.

D. The approach does not validate the use of the proposed technology for the application. You must either use an existing publicly documented XML schema or register the shared XML schema by using an authentication site.

Answer: B



Contact Test4pass

We are proud of our high-quality customer service, which serves you around the clock 24/7.

To get your problem resolved instantly, live support

Read Our Frequently Asked Questions (FAQs)

We have gathered the most frequently asked questions for you. Please read our list of FAQs.

Contact us by Live Messenger

Sales: [Test4pass\(at\)hotmail.com](mailto:Test4pass(at)hotmail.com)

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

Please allow up to 24 hours for us to respond

- MSN: Test4pass@hotmail.com