DUMPSARENA

GIAC GIAC Secure Software Programmer - C#.NET

GIAC GSSP-.NET

Version Demo

Total Demo Questions: 41

Total Premium Questions: 418

Buy Premium PDF

https://dumpsarena.co

sales@dumpsarena.co

sales@dumpsarena.co
dumpsarena.co

Topic Break Down

Topic	No. of Questions
Topic 1, Volume A	130
Topic 2, Volume B	99
Topic 3, Volume C	91
Topic 4, Volume D	78
Total	418

QUESTION NO: 1

Mario works as a Software Developer for BlueWell Inc. He develops an application, named App1, by using Visual Studio .NET. He deploys the application on a client computer. He wants to use the existing assembly named Assembly1. However, Assembly1 needs to be configured in the global assembly cache so that App1 is able to use it. He wants to ensure that the application data transferred to and from the server to the client computer is well secured. Which of the following tools will Mario use to accomplish the task?

A. Regasm.exe

B. Sn.exe

C. Mscorcfg.msc

D. Regsvcs.exe

ANSWER: C

Explanation:

Mscorcfg.msc is the correct historical .NET Framework administration tool for this scenario. The .NET Framework Configuration tool provided a graphical interface for configuring .NET Framework security policy and managing configuration-related settings for assemblies and applications. In the context of legacy Visual Studio .NET and .NET Framework deployments, it was used to configure trust and code-access-security policy so that an application could be granted the permissions required to load and use an assembly deployed for shared use. Such administration is relevant when an assembly is installed in the Global Assembly Cache, which enables strongly named assemblies to be shared by multiple applications on the machine. The GAC is a machine-wide assembly store and is specifically intended for assemblies that must be shared across applications; Microsoft describes its role and deployment considerations in its Global Assembly Cache documentation. The security-policy configuration capability addressed the question's legacy requirement to control how managed code executes and accesses protected resources on the client. This tool belongs to the older .NET Framework configuration model; contemporary .NET applications use different deployment and security mechanisms. Microsoft also documents the historical configuration-tool model in its .NET Framework configuration guidance.

QUESTION NO: 3

You work as a Software Developer for ABC Inc. The company uses Visual Studio .NET 2005 as its application development platform. You create an application named StringRW using .NET Framework. You store some characters into an array of Unicode characters. You need to write all or some of these characters into a String object. Which of the following code segments will you choose to accomplish this task?

Each correct answer represents a complete solution. Choose all that apply.

A. StringBuilder sb = new StringBuilder("This is my character String Builder"); char[] b = { 'c', 'l', 'a', 's', 's', 'r', 'i', 'g', 'h', 't', 'd', 'o', 'd' };
StringWriter sw = new StringWriter(sb); Object obj=(Object)b; sw.Write(obj);
Console.WriteLine(sb); sw.Close();

B. StringBuilder sb = new StringBuilder("This is my character String Builder"); char[] b = { 'c', 'l', 'a', 's', 's', 'r', 'i', 'g', 'h', 't', 'd', 'o', 'd' };
StringWriter sw = new StringWriter(sb); sw.Write(b);
Console.WriteLine(sb); sw.Close();

C. StringBuilder sb = new StringBuilder("This is my character String Builder"); char[] b = { 'c', 'l', 'a', 's', 's', 'r', 'i', 'g', 'h', 't', 'd', 'o', 'd' }; string str=new string(b);
StringWriter sw = new StringWriter(sb); sw.Write(str);
Console.WriteLine(sb); sw.Close();

D. StringBuilder sb = new StringBuilder("This is my character String Builder"); char[] b = { 'c', 'l', 'a', 's', 's', 'r', 'i', 'g', 'h', 't', 'd', 'o', 'd' };
StringWriter sw = new StringWriter(sb); sw.Write(b, 0, 5); Console.WriteLine(sb); sw.Close();

ANSWER: B C D

Explanation:

`sw.Write(b);` is correct because `StringWriter`, through its `TextWriter` implementation, provides a `Write(char[])` overload that writes every character in the supplied Unicode character array to the writer's underlying `StringBuilder`. The resulting builder therefore contains the original initial text followed by all characters from `b`.

`string str = new string(b); ... sw.Write(str);` is also a complete solution. The `String(char[])` constructor creates a `String` containing the characters from the array, and the `Write(string)` overload appends that string to the `StringWriter` output. This is appropriate when the character array should first be represented as a string value.

`sw.Write(b, 0, 5);` is correct when only part of the array is required. The `Write(char[], int, int)` overload writes a specified range, where `0` is the starting index and `5` is the number of characters to write. Thus, it appends the first five characters of the array. The relevant overloads are documented in TextWriter.Write Method, and the buffering behavior of the writer is documented in StringWriter Class.

QUESTION NO: 4

You work as a Web Application Developer for ABC Inc. The company uses Visual Studio .NET as its application development platform. You create a Web application using .NET Framework 2.0. You add a Web Form that contains a button named bCancel. The button allows users to exit the form. When users click the button, validation should not occur. However, during testing you find that clicking the button does not allow users to exit the form. You need to ensure that users can always exit the page. What will you do?

A. Set the CausesValidation property of the bCancel button to false.

B. Set the AccessibleName property of the validation controls on the Web Form to "false".

C. Set the Enabled property of the validation controls on the Web Form to false.

ANSWER: A

Explanation:

Set the CausesValidation property of the bCancel button to false. ASP.NET Web Forms controls that implement button behavior, including Button, can cause validation by default when clicked. If validators on the page detect invalid input, validation can prevent the normal postback action or server-side navigation logic associated with a cancel action from proceeding as intended. A cancel or exit button is specifically a case where the user should be able to leave the current operation without first satisfying required fields or other validation rules.

Setting `CausesValidation` to `false` on `bCancel` tells ASP.NET not to invoke the page's validation controls when that button initiates a postback. The button's click handler can then execute its exit behavior, such as redirecting to another page or returning to a previous workflow step, regardless of the current validity of user-entered data. This setting is scoped to the cancel button, preserving validation for submit or save controls that should enforce the form's input requirements. Microsoft documents `CausesValidation` as determining whether validation is performed when a button control is clicked; its default value is `true`. See Microsoft's Button.CausesValidation documentation and ASP.NET validation controls guidance.

QUESTION NO: 5

You work as a Software Developer for ABC Inc. The company uses Visual Studio.NET 2005 as its application development platform. You are developing an application using the

.NET Framework 2.0.

You are required to use a datatype that will store only numbers ranging from -32,768 to

32,767.

Which of the following datatypes will you use to accomplish the task?

Each correct answer represents a complete solution. Choose two.

A. int

B. string

C. short

D. System.Int16

ANSWER: C D

Explanation:

The correct types are `short` and `System.Int16`. In C#, `short` is the language keyword alias for the .NET `System.Int16` structure, so both names identify the same signed 16-bit integral type. A signed 16-bit value uses one bit for sign representation and has a defined inclusive range from −32,768 through 32,767, exactly matching the required range. Either form can be used in source code: developers commonly use `short` for concise C# declarations, while `System.Int16` may be used where the underlying Common Language Runtime type name is preferred or needed for clarity. The range is fixed by the type definition and does not vary between supported .NET platforms, making it appropriate when the application must represent only values within those boundaries. Microsoft's C# reference documents `short` as a 16-bit signed integer type and explicitly lists its minimum and maximum values. The .NET API documentation likewise identifies `System.Int16` as the signed 16-bit integer type. See C# integral numeric types and System.Int16 API documentation.

QUESTION NO: 6

You work as a .NET Trainer for ABC Inc. The Company uses .NET Framework as its application development platform. You are creating an application to demonstrate the use of datatypes. You create an unsigned Short datatype in the application. Which of the following values can be assigned to the unsigned Short datatype?

Each correct answer represents a complete solution. Choose three.

A. 123

B. 65,535

C. 127,625

D. 32,636

E. -123

ANSWER: A B D

Explanation:

In C#, the `ushort` keyword is an alias for `System.UInt16`, a 16-bit unsigned integral type. Being unsigned means that it represents whole-number values without a sign bit for negative numbers. Its valid inclusive range is 0 through 65,535. Therefore, `123` is valid because it is a nonnegative integer within the range. `65,535` is also valid because it is the maximum value supported by `UInt16`. `32,636` is valid as well, since it falls between zero and the maximum inclusive limit. These values can be represented directly by an unsigned 16-bit value and assigned to a `ushort` variable, subject to normal C# literal-conversion rules. Microsoft's documentation identifies `ushort` as the unsigned 16-bit integer type and specifies its range as 0 to 65,535. See Integral numeric types (C# reference) and System.UInt16 documentation.

QUESTION NO: 7

Allen works as a Software Developer for ABC Inc. He creates an application using Visual

Studio .NET 2005. He wants to ensure that the application must manage state effectively. State is how data is moved and persisted throughout the layers of the application. There are number of factors to validate state management. Which of the following factors will he consider to accomplish the task?

Each correct answer represents a part of the solution. Choose all that apply.

A. State persistence

B. Saving state

C. Caching

D. Shared state

E. View state

ANSWER: A B C D

Explanation:

Effective application state management requires deliberate decisions about where state resides, how long it remains valid, and how it is made available across application layers and requests. **State persistence** is essential because data that must survive beyond a single operation needs an appropriate durable store and lifecycle strategy. **Saving state** addresses the act of capturing relevant application or user data at defined points so it can be restored when the workflow requires it. **Caching** is a state-management consideration because temporarily retaining reusable data can reduce repeated computation or data access, provided expiration, invalidation, and consistency are defined. **Shared state** must also be considered when data is used by more than one component, request, process, or user context; its ownership, synchronization, scope, and concurrency requirements affect both correctness and security. Together, these concerns evaluate state's lifetime, storage, accessibility, and performance characteristics rather than treating state as an incidental implementation detail. Microsoft's guidance on ASP.NET state-management mechanisms describes the different scopes and persistence approaches available to applications, while its caching guidance emphasizes expiration and cache-data freshness as core design concerns. See ASP.NET State Management Overview and ASP.NET caching guidance.

QUESTION NO: 8

Which of the following provide simplified access to ASP.NET AJAX login, roles, and profile services from Windows Forms and WPF applications?

A. Entity Framework

B. AJAX application services

C. Client application services

D. Sync Framework

ANSWER: C

Explanation:

Client application services provide the intended bridge between Windows client applications and ASP.NET application-service functionality. In the .NET Framework, these services enable Windows Forms and WPF applications to use common membership-related capabilities—such as authenticating users, determining role membership, and retrieving or storing profile data—without requiring the application to implement all of the underlying service communication itself. The client application-services framework supplies client-side providers and configuration support that can connect to ASP.NET authentication, role, and profile services. This gives a desktop application a familiar .NET programming model while allowing identity and user-specific settings to be centrally managed by web-hosted application services. For example, a Windows client can use the client authentication service to log in against a configured ASP.NET service, then use role and profile providers as part of the same client-services design. Microsoft documents these capabilities under Client Application Services, including their use in Windows Forms and WPF applications. The associated web-side services are part of the ASP.NET application-services model described in ASP.NET Application Services Overview.

QUESTION NO: 9

Henry works as a Software Developer for SoftTech Inc. He creates a Windows form named

MyForm1. The form contains detailed information about a student. The form uses a ListBox control named ListBox1 that concatenates two strings displayed in two TextBox controls. Henry wants a method to return a value for the control. Which of the following options will he consider while creating a method for the control?

A. A method with a string return type.

B. A void method that passes only one parameter.

C. A method with an integer return type.

D. A void method that passes more than one parameter.

ANSWER: A

Explanation:

A method with a string return type. is appropriate because the intended result is formed by concatenating the text values from two TextBox controls. TextBox text is exposed through the `Text` property as a `string`, and concatenating two such values produces another `string`. A helper method can therefore obtain the two text values, combine them, and return the completed textual value to the caller. The caller can then use that returned value when adding an item to `ListBox1.Items`, displaying it elsewhere, or performing any further validation or formatting. In C#, a non-`void` method declares the type of value it will return, and each reachable completion path must return a value compatible with that declared type. For example, the method could use a signature such as `string GetStudentText()` and return `textBox1.Text + textBox2.Text`. This directly represents the data type being created and makes the method's contract clear: it provides a textual result rather than merely carrying out an action. Microsoft's documentation on C# method return values describes this typed return-value contract, while the String.Concat documentation confirms that combining strings results in a string.

QUESTION NO: 10

You work as a Software Developer for ABC Inc. The company uses Visual Studio.NET

2005 as its application development platform. You are creating an application using .NET Framework 2.0. You are using regular expression in the application to validate email, phone number, etc. You are required to match any single character except "\n". Which of the following wildcard characters will you use to accomplish the task?

A. x/y

B. +

C. ?

D. .

ANSWER: D

Explanation:

The correct wildcard character is **.** (the period). In the .NET regular expression engine, a period is a metacharacter that matches one arbitrary character, except the newline character `\n` by default. This makes it useful when a pattern must accept any single ordinary character in a value being validated, while not allowing the match to span onto a new line. For example, the expression `a.c` matches strings such as `abc` and `a-c`, because the period represents exactly one character between `a` and `c`.

This default behavior can be changed by enabling single-line mode, such as with `RegexOptions.Singleline` or the inline option `(?s)`. In single-line mode, the period also matches newline characters. Since the requirement specifically excludes `\n`, the normal period metacharacter without single-line mode is the appropriate choice. Microsoft documents this behavior in its discussion of regular-expression character matching and options: Character classes in regular expressions and Single-line mode.

QUESTION NO: 11

You work as a Software Developer for ABC Inc. The company uses Visual Studio.NET 2005 as its application development platform. You create a remoting application that provides stock information to customers using .NET Framework. The server component raises an event on the client computer when certain conditions are met. You must ensure that the server raises exactly one event for each client application that is registered for the event. What will you do to accomplish this task?

A. Configure the server class as a SingleCall Activated Object (SAO) and check for duplicate client delegate methods before raising the event.

B. Configure the server class as a Singleton Activated Object (SAO) and check for duplicate client delegate methods before raising the event.

C. Configure the server class as a Client Activated Object (CAO) and override the CreateObjRef method to check for duplicate client delegate methods before raising the event.

D. Configure the server class as a Client Activated Object (CAO) and check for duplicate client delegate methods before raising the event.

ANSWER: B

Explanation:

Configure the server class as a Singleton Activated Object (SAO) and check for duplicate client delegate methods before raising the event. A singleton server-activated remoting object provides one shared server instance for all connected clients. Consequently, that single instance owns a unified event subscription list and can coordinate notifications across every registered client application. This shared state is essential when the server must control delivery behavior consistently rather than having separate per-call server instances with isolated state.

Before publishing the stock-condition notification, the server should inspect or maintain its registered callback delegates so that a client delegate is represented only once. The server can then invoke each distinct registered callback exactly once, meeting the requirement even when a client attempts to register the same handler more than once. .NET delegates represent invocation targets and can contain an invocation list, making subscription management a necessary part of reliable event delivery. Microsoft's remoting activation guidance describes singleton server activation as a shared object instance, while its delegate documentation explains delegate invocation behavior: .NET remoting activation and Delegate.GetInvocationList.

QUESTION NO: 12

Maria works as a Software Developer for BlueWell Inc. She develops an application, named App1, using Visual C# .NET. The application displays employee details from a SQL Server database. Maria wants to use a string array, named MyArray, in the application code to store employee names. Which of the following statements will she use to declare

MyArray?

A. Option Base 1 string[] MyArray = new string[9];

B. string[] MyArray = new string[9];

C. Option Base 0 string[] MyArray = new string[9];

D. string[] MyArray = new string[0 to 9];

E. string MyArray[9] = new string;

ANSWER: B

Explanation:

`string[] MyArray = new string[9];` is the correct C# declaration and initialization for an array intended to hold employee names. The `string[]` type specifies that `MyArray` is a one-dimensional array whose elements are .NET `String` references. The `new string[9]` expression allocates the array with a fixed length of nine elements. Because

`string` is a reference type, each newly allocated element is initially `null` until the application assigns an employee name to that position. For example, the code can subsequently assign a value with `MyArray[0] = "Maria";`. C# arrays use zero-based indexing, so an array created with length nine provides valid positions from index 0 through index 8. This syntax combines the type declaration, variable declaration, and array allocation in a standard C# statement and is appropriate when the number of required name entries is known at allocation time. Microsoft's C# array documentation describes creating arrays through `new` and specifies that the length expression determines the number of elements: C# arrays. The behavior of unassigned reference-type array elements is also consistent with .NET's default-value rules.

QUESTION NO: 13

Charles works as a Web Developer for Cyber Net Inc. The company frequently receives production information from external vendors in the form of XML data. Charles stores these XML data in a disk container of the Web server as .xsd schema files and .xdr schema files. There are specific rules for writing XML codes to define the structure and data types for XML documents. Which of the following criteria should NOT be considered while writing an

XML document?

Each correct answer represents a complete solution. Choose all that apply.

A. A well-formed XML document should contain a start tag but not necessarily an end tag.

B. A well-formed XML document should contain a start tag and an end tag.

C. An XML document should be specified according to the application or business requirement s.

D. An XML document should have predefined elements and attributes.

ANSWER: A D

Explanation:

A well-formed XML document should contain a start tag but not necessarily an end tag should not be considered because XML element tags must be properly closed. In ordinary elements, the end tag must match the start tag, such as `<item>...</item>`. When an element has no content, XML permits the equivalent self-closing syntax, such as `<item />`; this is still a complete closing construct, not an unclosed start tag. XML's well-formedness rules require proper nesting and matching tags so that parsers can unambiguously identify the document's tree structure. The XML specification describes these requirements in its discussion of elements and matching end-tags: W3C Extensible Markup Language (XML).

An XML document should have predefined elements and attributes also should not be considered a universal requirement. XML itself defines syntax for marking up structured data, but it does not impose a fixed vocabulary of element or attribute names. A document may use names designed for its business purpose, provided the resulting document is well formed. An XSD schema can optionally define permitted elements, attributes, data types, and structural constraints when an organization needs validation or an agreed contract with vendors. This distinction between well-formed XML and schema-valid XML is reflected in Microsoft's XML Schema validation documentation.

QUESTION NO: 14

All of the following are benefits for query expressions allowed by Language-integrated query except for which one?

A. Rich metadata

B. Compile-time syntax checking

C. Static typing and IntelliSense

D. Imperative code

ANSWER: D

Explanation:

Imperative code is the correct exception because LINQ query expressions are intended to provide a declarative way to state what data is wanted, rather than requiring the programmer to specify the step-by-step control flow used to obtain it. In a query expression, the developer describes operations such as filtering, ordering, grouping, joining, and projecting data through a consistent C# syntax. The compiler translates that expression into the appropriate method calls or provider-specific representation.

This language integration gives queries first-class treatment within C#. Query syntax is parsed and checked by the compiler, works with the type system, and is supported by editor tooling such as IntelliSense. It can also operate over in-memory collections and, through query providers, other data sources. Although query execution may ultimately involve procedural operations internally, writing imperative code is not itself a benefit delivered by query-expression syntax. The important benefit is that the query can be expressed declaratively while retaining normal C# compile-time support and tooling. Microsoft describes LINQ as integrating query capabilities directly into C# and explains that query expressions are translated into standard query operator method calls. See Language Integrated Query (LINQ) and Query expression basics.

QUESTION NO: 15

You work as a Web Application Developer for ABC Inc. The company uses Visual Studio .NET as its application development platform. You are creating an ASP.NET Web application using the .NET Framework. The application will allow users to post and reply to messages after logging. You create the authentication mechanisms for the application. You use SQL Server 2005 and the Active Directory service interchangeably for authentication. You are required to provide a method for users of the application to register. You also want to ensure that the application utilizes multiple authentication methods when users register.

What will you do?

A. Create a form that contains a CreateUserWizard control and configure custom membership providers.

B. Create a form that contains a CreateUserWizard control and configure Membership providers.

C. Create a Web custom component and Configure Membership Providers.

D. Create a Web custom component that offers a new user registration form and stores the outcome in XML that matches a defined schema.

ANSWER: B

Explanation:

Create a form that contains a CreateUserWizard control and configure Membership providers is correct because ASP.NET Membership separates user-registration functionality from the underlying identity store through a provider model. The `CreateUserWizard` control supplies a ready-made registration workflow, including account-creation fields, validation, and calls to the configured membership service. Membership providers are registered in the application configuration and can be selected for the relevant registration workflow through the control's membership-provider settings.

This approach lets the application work with more than one membership implementation rather than embedding storage-specific registration logic in the page. The SQL membership provider can support SQL Server-backed accounts, while an appropriate membership-provider implementation can integrate another identity source such as Active Directory. The provider abstraction gives the application a consistent registration interface while allowing the configured provider to perform the actual credential and account operations for its backing store. This is the intended ASP.NET design for reusable, configurable authentication and membership services.

See Microsoft documentation for CreateUserWizard and ASP.NET Membership Providers documentation.

QUESTION NO: 17

You work as a Software Developer for ABC Inc. The company uses Visual Studio.NET 2005 as its application development platform. You are developing an application that will manage rental properties for a property management agency. When a property is rented, a lease agreement is created. At that time a new lease is entered into the application, the starting date of the lease may not yet be known.

You need to write code that stores the lease start date entered by a user in a variable. The variable that stores the lease start date should contain no value, by default. If no lease start date is entered, the variable should have no value assigned to it.

Which code segment should you choose?

A. DateTime? leaseStartDate = null;

B. DateTime leaseStartDate = null;

C. DateTime? leaseStartDate = new DateTime();

D. DateTime leaseStartDate = new DateTime();

ANSWER: A

Explanation:

`DateTime? leaseStartDate = null;` is correct because the nullable form of `DateTime` represents a date and time that may either hold a valid `DateTime` value or have no value. The `?` syntax is shorthand for `Nullable<DateTime>`, a value-type wrapper provided by C#. Assigning `null` establishes the required initial state: no lease start date has been supplied. When the user later supplies a date, the application can assign that `DateTime` value directly to the nullable variable. The program can then safely determine whether a date was provided through `HasValue`, or retrieve it only when present through `Value` or a null-coalescing fallback. This accurately models the business condition that a lease start date is optional at the time the lease record is created, while still retaining the type safety and date operations of `DateTime` when a value exists. Nullable value types were introduced in C# 2.0, making this syntax appropriate for the Visual Studio 2005/C# environment described. Microsoft documents nullable value types and their null state at Nullable value types and documents the underlying System.Nullable<T> structure.

QUESTION NO: 18

You work as a Software Developer for ABC Inc. You use C# .NET to develop a windows application. The application will implement a role-based authorization scheme that is based on a Microsoft SQL Server database of user names. Users will enter their names in a text box named UserName. You must ensure that users are assigned the supervisor role and the PG role by default. Which of the following code segments will you use to accomplish this task?

A. GenericIdentity identity = new GenericIdentity(UserName.Text); string[] RoleArray = {"Supervisor", "PG"};
WindowsPrincipal principal = new WindowsPrincipal(identity);

B. WindowsIdentity identity = new WindowsIdentity.GetAnonymous(); string[] RoleArray = {"Supervisor", "PG"};
WindowsPrincipal principal = new GenericPrincipal(identity, RoleArray);

C. GenericIdentity identity = new GenericIdentity(UserName.Text); string[] RoleArray = {"Supervisor", "PG"};
GenericPrincipal principal = new GenericPrincipal(identity, RoleArray);

D. WindowsIdentity identity = new WindowsIdentity.GetCurrent(); string[] RoleArray ={"Supervisor", "PG"};
GenericPrincipal principal = new GenericPrincipal(identity, RoleArray);

ANSWER: C

Explanation:

`GenericIdentity identity = new GenericIdentity(UserName.Text); string[] RoleArray = {"Supervisor", "PG"}; <br />GenericPrincipal principal = new GenericPrincipal(identity, RoleArray);` is correct because it creates an application-defined identity using the user name supplied in `UserName.Text`, then associates that identity with the required application roles through a `GenericPrincipal`. `GenericIdentity` is intended for scenarios in which the application manages the identity rather than relying on an underlying Windows account. Its name value can represent the authenticated user name retrieved or validated against the SQL Server user store.

The role array passed to the `GenericPrincipal` constructor establishes the roles associated with that identity. Consequently, role checks such as `principal.IsInRole("Supervisor")` and `principal.IsInRole("PG")`

evaluate successfully for this principal, meeting the requirement to assign both roles by default. In a complete application, the resulting principal would normally be assigned to the executing context, such as through `Thread.CurrentPrincipal`, so authorization code uses it consistently. Microsoft documents that `GenericPrincipal` represents a generic principal with an identity and an array of roles, and that `GenericIdentity` represents a generic identity. See Microsoft's GenericPrincipal documentation and Microsoft's GenericIdentity documentation.

QUESTION NO: 19

ASP.NET version 2.0 health monitoring supports an _____.

A. Event driven model

B. Event provider model

C. Event log model

D. Event API model

ANSWER: B

Explanation:

Event provider model is correct because ASP.NET health monitoring is designed as an extensible provider-based system. Applications raise health-monitoring events for significant operational and security-relevant conditions, such as authentication activity, application lifetime events, errors, or request-processing conditions. Configured health-monitoring providers receive those events and deliver their information to a selected destination. This architecture lets an application use built-in delivery mechanisms or implement a custom provider when logging, alerting, storage, or integration requirements differ from the defaults.

The configuration associates event types with providers through health-monitoring rules, allowing developers and administrators to decide which events should be captured, where they should be sent, and under what conditions they should be recorded. The provider abstraction is the key feature: event generation is separated from event consumption and delivery. This makes monitoring configurable without requiring application code to be rewritten merely to change the event destination or reporting behavior. Microsoft describes ASP.NET health monitoring as an infrastructure based on events, providers, and rules, and documents the associated management classes in the `System.Web.Management` namespace. See the ASP.NET Health Monitoring Overview and the System.Web.Management namespace documentation.

QUESTION NO: 20

You work as a Software Developer for ABC Inc. The company uses Visual Studio .NET 2008 as its application development platform. You create an ASP.NET Web application using .NET Framework 3.5. The application stores sensitive profile data in a SQL Server 2008 database. You are required to make sure that no profile data is stored in clear text.

What will you do to accomplish the task?

A. Use the Aspnet_regiis.exe tool to encrypt the connection string that is used to connect to the database.

B. Create a custom profile provider and ensure that the provided information is encrypted before it is stored in the database.

C. Enable the secure sockets layer encryption for the SQL Server connections.

D. Use SQL Profiler.

ANSWER: B

Explanation:

Create a custom profile provider and ensure that the provided information is encrypted before it is stored in the database. This directly protects the sensitive profile values at rest, which is the stated requirement. ASP.NET profile data is managed through a provider model: a custom provider can control how profile properties are serialized, protected, written to storage,

read back, and decrypted for authorized application use. The provider should apply authenticated encryption before persistence, with encryption keys managed separately from the database and access restricted according to least-privilege principles. This design ensures that database rows and backups do not contain usable clear-text profile values if they are viewed outside the application's authorized processing path. Microsoft's ASP.NET profile-provider documentation describes the provider-based architecture and the ability to implement custom profile storage behavior. For the cryptographic implementation, use established .NET cryptographic APIs and an authenticated encryption design rather than a custom encryption algorithm. See ASP.NET Profile Providers and .NET cryptographic services.

QUESTION NO: 21

You work as a Software Developer for ABC Inc. You create a Web service named MyWebService using Visual Studio .NET. You implement a public Web method. You make the method accessible through a Web service application by attaching the WebMethod attribute to the public method. The public method is stored in the class file of the Web service file. When you create a Web service project using the Web service project template, by default, one public method named HelloWorld is created. The HelloWorld public method does not become available through the Web service until you attach the WebMethod attribute. The method must be applied to each method declaration. Now, you want to configure a Web service method. You want to ensure that the Web method performs updates on more than one databases or if the sequence of updates is critical to the overall success of the Web method. Which of the following properties will you use to accomplish this task?

A. TransactionOption

B. Description

C. BufferResponse

D. EnableSession

E. Set TransactionOption to Required.

ANSWER: A E

Explanation:

`TransactionOption` is the `WebMethod` attribute property used to request transactional behavior for an ASP.NET XML Web service method. A transaction establishes an atomic unit of work: when the method updates multiple transactional resources, such as databases, the work can be committed together only after the method completes successfully. This is appropriate when the updates must remain consistent or their order is essential to the operation's overall success. The `TransactionOption` value is selected from `System.EnterpriseServices.TransactionOption`. Setting it to `Required` instructs the method to participate in an existing transaction when one is available, or to create a transaction when one is not available. Thus, an implementation can declare the behavior directly on the Web method, for example: `[WebMethod(TransactionOption = TransactionOption.Required)]`. The participating databases and resource managers must support the relevant transaction-coordination infrastructure for a distributed transaction to provide atomicity across them. Microsoft documents `WebMethodAttribute.TransactionOption` as the transaction option applied to an XML Web service method, and documents `Required` as joining an existing transaction or starting a new one when necessary. See WebMethodAttribute.TransactionOption and TransactionOption enum.

QUESTION NO: 22

You work as a Web Application Developer for SunInfo Inc. The company uses Visual

Studio 2008 as its application development platform. You create a Web application using .NET Framework 3.5. You want to implement state management in your application. In which of the following stages of the application life cycle will your application actually start?

A. On creation of an instance of the HttpApplication class

B. When an instance of the ApplicationManager class is created

C. When instances of the core objects such as HTTPContext, HTTPRequest, and HTTPResponse are created

D. When a user first makes a request for a page in your Web site

ANSWER: A

Explanation:

On creation of an instance of the HttpApplication class is correct. In the classic ASP.NET application life cycle, ASP.NET starts the application by creating an application domain and initializing an `HttpApplication` instance for the application. This initialization establishes the application-level processing pipeline and raises application-start processing, including the `Application_Start` event when it is defined in `Global.asax`. It is at this point that application-scoped initialization is performed, such as configuring application-wide resources or initializing state-management infrastructure that will be shared across requests. ASP.NET can create multiple `HttpApplication` instances to process concurrent requests, but application startup initialization is associated with the first instance created for the application domain. The external request is what causes ASP.NET to load and initialize the application when it is not already running; however, the life-cycle start stage itself is identified by creation of the `HttpApplication` instance. See Microsoft's documentation for the HttpApplication class and the ASP.NET application life cycle overview.

QUESTION NO: 23

You work as a Web Application Developer for SunInfo Inc. The company uses Visual

Studio 2008 as its application development platform. You create a Web application using

.NET Framework 3.5. You create a data-bound control. You want to retrieve data from a relational database and edit data on the Web page by using the data-bound control. Which of the following data source classes will you use to accomplish the task?

A. ObjectDataSource

B. LinqDataSource

C. SqlDataSource

D. XmlDataSource

ANSWER: C

Explanation:

SqlDataSource is the appropriate data-source control because it is designed to connect ASP.NET data-bound controls directly to relational databases through an ADO.NET provider, such as SQL Server. It can supply records through a `SelectCommand` and supports database modification through configured `UpdateCommand`, `InsertCommand`, and `DeleteCommand` properties. This makes it suitable for a page where a bound control, such as GridView, displays relational data and allows a user to edit that data in the browser.

For editing to work, the bound control enables its editing capability and the SqlDataSource is configured with the required update operation and parameters. ASP.NET then coordinates the control's edit values with the data source's update command when the user saves the change. SqlDataSource also supports connection strings stored in configuration, parameterized commands, and provider-independent database connectivity, all of which fit the stated ASP.NET Web Forms and .NET Framework 3.5 scenario. Microsoft's documentation describes SqlDataSource as a control for working with SQL data and performing data retrieval and modification operations. See Microsoft Learn: SqlDataSource Class and Microsoft Learn: Editing Data with the SqlDataSource Control.

QUESTION NO: 24

You work as a Software Developer for InfoTech Inc. You develop an application named MyApp that works as a scientific calculator. A user inputs data and retrieves the calculated results based on the input values. The application uses a SQL database that contains two tables. One table contains data related to standard calculation specifications and the other table stores input values entered by the user. You want to ensure that the user is able to retrieve appropriate data fast. Which of the following actions will you take to accomplish the task?

Each correct answer represents a part of the solution. Choose all that apply.

A. Use more than one component to process input data.

B. Write code that performs multiple tasks simultaneously.

C. Run multiple processes on a single thread.

D. Use a single component to process input data.

E. Run multiple processes on separate threads.

ANSWER: B D E

Explanation:

"Write code that performs multiple tasks simultaneously," "Use a single component to process input data," and "Run multiple processes on separate threads" support a responsive design for a calculator application that must obtain and process related database information quickly. Independent work, such as retrieving calculation specifications and processing user-supplied values, can be scheduled concurrently rather than forcing the user interface or request-processing path to wait for each operation serially. In .NET, task-based asynchronous programming provides a structured approach for coordinating concurrent operations while preserving responsiveness; when the work is I/O-bound, asynchronous database calls can release the calling thread while the database operation is pending. Microsoft documents this model in its asynchronous programming with async and await guidance.

Using a single component to process the input-data workflow also centralizes validation, calculation orchestration, and data-access coordination. This reduces unnecessary intercomponent communication and duplicated processing in the latency-sensitive path, while separate worker threads or asynchronous tasks handle independent operations. Concurrency should be bounded and coordinated so that it improves throughput and responsiveness without creating unsafe shared-state access. The Task Parallel Library provides the .NET abstractions used to run and coordinate concurrent work, as described in Microsoft's Task Parallel Library documentation.

QUESTION NO: 25

Sandra works as a Software Developer for ABC Inc. She develops a Windows form named MyForm1. She wants to add a Button control named button1 to a Panel control named panel1 at runtime. Therefore, she writes the following code:

```
1. private void MyForm1_Load(object sender, EventArgs e)
2. {
3. Button button1 = new Button();
4. button1.Location = new Point(50,50);
5. button1.Size = new Size(100, 30);
6. button1.Text = "My First button";
7. Panel panel1 = new Panel();
8. panel1.Location = new Point(10, 100);
9. // Write code here
10. // Write code here
11. }
```

Which of the following code statements will Sandra write in line numbers 9 and 10? (Line numbers are for reference only.)

Each correct answer represents a part of the solution. Choose two.

A. She will write the following code statement in line number 9: Form1.Controls.Add(button1);

B. She will write the following code statement in line number 9:
this.Controls.Add(button1);

C. She will write the following code statement in line number 9:
this.Controls.Add(panel1);

D. She will write the following code statement in line number 10:
button1.Controls.Add(panel1);

E. She will write the following code statement in line number 10:
panel1.Controls.Add(button1);

ANSWER: C E

Explanation:

The required control hierarchy is the form containing `panel1`, with `panel1` containing `button1`. Therefore, `this.Controls.Add(panel1);` adds the dynamically created panel to the current form's `Controls` collection. Once it has been added, the panel becomes part of the form's visual control tree and its location is interpreted relative to the form's client area.

`panel1.Controls.Add(button1);` then assigns the dynamically created button to the panel's child-control collection. This is what makes the button a child of the panel rather than a direct child of the form. Consequently, the button's `Location` coordinates are interpreted relative to the panel's client area, which is the intended behavior when placing a control inside a container at runtime.

Windows Forms containers expose a `Controls` collection for managing child controls, and the collection's `Add` method establishes this parent-child relationship. The ordering shown also constructs the intended hierarchy explicitly: add the container to the form and add the contained button to that container. See Control.Controls Property and Control.ControlCollection.Add Method.

QUESTION NO: 27

Charles works as a Web Developer for TechNet Inc. He develops an application named AdAgentsApp for advertising agents by using Visual Studio .NET. AdAgentsApp uses several Web services provided by the company. The employees in the IT department use only the existing port numbers 80 (HTTP protocol) and 443 (HTTPS protocol) for firewall security. However, they are forbidden to open any other new ports. These ports have been used to ensure the highest security level of authentication.

Which of the following actions should Charles take to maintain the highest security?

Each correct answer represents a part of the solution. Choose two.

A. Use Windows Basic Authentication with SSL.

B. Use Windows Digest Authentication.

C. Use Custom Authentication.

D. Send the encrypted information in SOAP headers.

E. Use Windows Authentication with Client Certificates.

ANSWER: C D

Explanation:

Custom Authentication is appropriate because it lets the Web service implement an authentication scheme tailored to the organization's security requirements without requiring a separate network listener or newly opened firewall port. The service can validate application-specific credentials or security tokens, establish the caller's identity, and apply authorization decisions while the request is delivered through the already permitted HTTP or HTTPS endpoints. A secure custom design

should validate credentials and tokens server-side, use expiration and replay protections where applicable, and protect all credential-bearing traffic with HTTPS.

Sending encrypted information in SOAP headers provides message-level protection for sensitive authentication or authorization data. Unlike transport-only protection, SOAP security headers can protect selected message elements and preserve their confidentiality when a message passes through intermediaries or is processed beyond the initial transport connection. This approach is particularly relevant to Web services because identity assertions, tokens, and related security material can travel with the SOAP message and be verified by the receiving service. Using HTTPS on port 443 together with properly encrypted SOAP security headers provides layered protection while remaining within the organization's allowed ports. Microsoft's guidance on SOAP message security is available in Message Security with a Windows Client, and its authentication overview is available in Authentication in WCF.

QUESTION NO: 28

You work as a Software Developer for ABC Inc. The company uses Visual Studio.NET

2005 as its application development platform. You create a Web service application using .NET Framework. The Web service provides confidential data of employees to applications that manage access to company facilities. The Web service is accessible by using TCP and is sheltered by using WSE 3.0. The company has implemented fingerprint readers to grant employees access to the facilities. All the captured images of the employees' fingerprints are retained by the Web service application. You must make certain that the existing WSE encryption policy can be applied to the fingerprint image. You are required to provide the solution that must reduce the size of the Web service message.

What will you do to accomplish this task?

A. Configure the Web service to use base64 encoding to pass the binary fingerprint image.

B. Create a SOAP extension to manage encryption for the message.

C. Configure the Web service to use Message Transmission Optimization Mechanism to pass the binary fingerprint image.

D. Create a SOAP filter to manage encryption for the message.

ANSWER: C

Explanation:

Configure the Web service to use Message Transmission Optimization Mechanism to pass the binary fingerprint image. MTOM is designed for SOAP services that must carry substantial binary content, such as fingerprint images, while retaining the SOAP/XML message model required by Web-service security policies. Instead of placing the complete image directly into the SOAP XML as Base64 text, MTOM represents the binary value in the SOAP infoset and transmits the actual bytes as an optimized MIME part. This avoids the approximately one-third size expansion normally associated with Base64 encoding and therefore reduces message size and network overhead. The SOAP message still logically contains the binary value, allowing the established WSE security processing and encryption policy to protect the message content consistently. This is particularly important for biometric data, which is sensitive and should remain protected in transit without sacrificing efficient transport. The W3C MTOM specification defines this optimized serialization approach for SOAP messages containing binary data: W3C SOAP Message Transmission Optimization Mechanism. Microsoft also documents that MTOM encoding optimizes the transmission of binary data in SOAP messages: Microsoft Learn: Large Data and Streaming.

QUESTION NO: 29

You work as a Software Developer for ManSoft Inc. You use Microsoft Visual Studio to create a Web service named MyWebService. You create a SOAP message that is not secure in the Web service. You want to use the ReceiveSecurityFilter class in the Web service to handle the receiving and securing of SOAP messages. Which of the following code segments will you use to accomplish the task?

Each correct answer represents a part of the solution. Choose all that apply.

A. public class MyReceiveSecurityFilter : ReceiveSecurityFilter
{
public MyReceiveSecurityFilter(string service, Boolean client) : base(service, client)

{ }
public override void ValidateMessageSecurity(SoapEnvelope envelope, Security security)
{
// Implementation validation here
}
}

B. public class MyReceiveSecurityFilter : ReceiveSecurityFilter
{
public MyReceiveSecurityFilter(string service, Boolean client) : base(service, client)
{ }
public override void ValidateMessageSecurity(SoapEnvelope envelope, Security security)
{
// Implementation validation here
}
}

C. public class MyReceiveSecurityFilter : ReceiveSecurityFilter
{
public MyReceiveSecurityFilter(string service, Boolean client)
{ }
public override void ValidateMessageSecurity(SoapEnvelope envelope, Security security)
{
// Implementation validation here
}
}

D. public class MyReceiveSecurityFilter : ReceiveSecurityFilter
{
public MyReceiveSecurityFilter(string service, Boolean client) : base(service, client)
{ }
public void ValidateMessageSecurity(SoapEnvelope envelope, Security security)
{
// Implementation validation here
}
}

ANSWER: A B

Explanation:

A class derived from `ReceiveSecurityFilter` must initialize its base class by calling the inherited constructor with the service and client arguments. In C#, this is done through a constructor initializer, written as `: base(service, client)`, which executes before the derived constructor body. This supplies the configuration information required by the receive-side SOAP security filter. The derived filter must also override `ValidateMessageSecurity(SoapEnvelope envelope, Security security)`. Declaring this member with `override` ensures that the framework invokes the custom validation implementation as part of processing an incoming SOAP message. The method receives both the SOAP envelope and its associated security data, allowing the service to apply the intended message-security validation logic. Both listed correct code segments use the required constructor initializer and provide the required overridden validation method; they are equivalent implementations. Microsoft's C# documentation describes how base-class constructor initializers work at Constructors (C# Programming Guide). The requirement that an overridden member explicitly use `override` is described in The override keyword.

QUESTION NO: 30

You work as a Software Developer for ABC Inc. The company uses Visual Studio.NET 2005 as its application development platform. You are creating an application that reads data from and writes data to a file using the .NET Framework 2.0. You want to restrict users to read characters from a byte stream and write characters to a stream. Which of the following classes will you use to accomplish the task?

Each correct answer represents a part of the solution. Choose two.

A. StringReader

B. StringWriter

C. StreamReader

D. StreamWriter

ANSWER: C D

Explanation:

`StreamReader` and `StreamWriter` are the appropriate character-oriented stream classes for this requirement. `StreamReader` reads characters from an underlying byte stream by decoding bytes with a character encoding, such as UTF-8. This allows an application to consume text from a file or another stream source using character-based operations such as `Read`, `ReadLine`, and `ReadToEnd`. The underlying source can be supplied directly as a stream or opened by specifying a file path.

`StreamWriter` performs the corresponding output role: it encodes characters and writes the resulting bytes to an underlying stream. It supports text-oriented operations such as `Write` and `WriteLine`, making it suitable for creating or updating text-file content. Together, these classes provide the standard .NET pattern for reading and writing textual data while the file system and base stream remain byte-based. Their encoding behavior is particularly important because it ensures that characters are converted to and from bytes consistently. See the StreamReader documentation and the StreamWriter documentation.

QUESTION NO: 31

You work as a Software Developer for ABC Inc. The company uses Visual Studio.NET as its application development platform. You create an application using the .NET Framework. You need to use the regular expressions to provide a concise and flexible means for identifying strings of text of interest, such as particular characters, words, or patterns of characters. Which operation will you use to construct the regular expressions?

Each correct answer represents a complete solution. Choose all that apply.

A. Quantification

B. Grouping

C. Alternation

D. Phishing

ANSWER: A B C

Explanation:

Regular-expression construction uses composable language features to describe the required text pattern. **Grouping** encloses a subexpression, typically with parentheses, so several tokens can be treated as one unit. This lets a pattern apply subsequent matching logic to an entire word fragment or repeated sequence and can capture the matched value for later use. **Quantification** specifies how many times the preceding character, character class, group, or other expression may occur. Common quantifiers include `*`, `+`, `?`, and brace ranges such as `{2,5}`; these allow one pattern to match variable-length input without manually listing every possible length. **Alternation** uses the pipe metacharacter, `|`, to match one of multiple expressions, such as a choice among permitted words or formats. Together, these operations provide the concise and flexible pattern-building capabilities requested: grouping defines units, quantification controls repetition, and alternation expresses choices. The .NET regular-expression language documents grouping constructs, quantifiers, and alternation as core elements of regex syntax. See .NET regular expression language quick reference and Microsoft documentation on grouping constructs.

QUESTION NO: 32

Daniel works as a Software Developer for GenTech Inc. He develops an ASP.NET application, named

App1, using Visual Studio .NET. He creates an interface in App1, named Employees. The Employees interface contains declaration for a method, named Method1. Daniel creates another interface, named HeadOfDept. HeadOfDept inherits from the Employees interface and contains the declaration for a method, named Method2. He then creates a class named Manager as follows:

public class Manager : HeadOfDept

Which of the following statements about the application are true?

Each correct answer represents a complete solution. Choose all that apply.

A. The Manager class must implement the Method1 method.

B. The Manager class implements the HeadOfDept interface.

C. The Manager class must implement the Method2 method.

D. The Method1 method can have any access modifier.

ANSWER: A B C

Explanation:

When a class declares an interface after the colon in its class declaration, it implements that interface. Therefore, `Manager` implements `HeadOfDept`. Interface inheritance is transitive: because `HeadOfDept` inherits from `Employees`, the contract represented by `HeadOfDept` includes both its own members and the members inherited from `Employees`. Consequently, a non-abstract `Manager` class must provide implementations for both `Method1` and `Method2`, unless those members are implemented through valid explicit interface implementations. Under the ordinary implicit implementation form, the implementing methods must be public because interface members are publicly accessible through an interface reference. This preserves the interface contract for callers using either a `HeadOfDept` reference or an `Employees` reference to a `Manager` instance. Microsoft's documentation explains that a class implementing an interface must implement the required members, including inherited interface members, and describes the accessibility rules for implicit versus explicit interface implementation. See C# interfaces and explicit interface implementation.

QUESTION NO: 33

You work as a Software Developer for ABC Inc. The company uses Visual Studio .NET 2008 as its application development platform. You create an ASP.NET Web application using .NET Framework 3.5. You create your own HTTP handlers that render custom output to the browser in the application. Other than this, under which of the following operations will you use your HTTP handlers?

Each correct answer represents a complete solution. Choose two.

A. Image server

B. Aggregator

C. Image mapping

D. RSS feeds

ANSWER: A D

Explanation:

HTTP handlers are appropriate for an **Image server** because a handler can receive a request for an image, perform authorization or lookup logic, retrieve image bytes from a database or other storage, set the appropriate HTTP content type, and write the binary response directly to the output stream. This provides fine-grained control over image delivery without requiring an .aspx page and its page-processing lifecycle.

RSS feeds are also a strong use case because a handler can generate and return XML in response to a feed URL. The handler can query current content, serialize it into RSS-compliant XML, set an XML content type, and efficiently send the resulting document to feed readers and browsers. ASP.NET routes requests to a registered handler based on configured mappings, making handlers useful for resource-style endpoints that need custom request processing and direct response generation.

The `IHttpHandler` interface defines the request-processing contract used by ASP.NET handlers, including the `ProcessRequest` method through which the response is produced. See Microsoft's IHttpHandler documentation and Microsoft's overview of ASP.NET HTTP handlers.

QUESTION NO: 34

Peter works as a Software Developer for PentaSoft Inc. He develops an application, named App1, using Visual C# .NET. The application is intended to deal with several file input and output operations. He uses the following try structure in the application code to handle errors that occur during the execution of App1.

```
try
{
//Statements that might cause a runtime error.
}
catch (System.IO.FileNotFoundException ex)
{
//Statements
}
catch (System.IO.PathTooLongException ex)
{
//Statements
}
catch (System.IO.IOException ex)
{
//Statements
}
catch (Exception ex)
{
//Statements
}
```

What will happen if an exception of type System.IO.DirectoryNotFoundException occurs in the try block?

A. The catch block that handles an exception of type IOException will be executed.

B. The catch block that handles an exception of type Exception will be executed.

C. The catch block that handles an exception of type FileNotFoundException will be executed.

D. The catch block that handles an exception of type PathTooLongException will be executed.

ANSWER: A

Explanation:

The catch block that handles an exception of type IOException will be executed. `System.IO.DirectoryNotFoundException` derives from `System.IO.IOException`, so an instance of `DirectoryNotFoundException` is compatible with a `catch (System.IO.IOException ex)` handler. C# evaluates catch clauses in source order and selects the first handler whose declared exception type can handle the exception object thrown. In this sequence, neither of the preceding file-specific handlers is the matching handler for a directory-not-found condition. Execution therefore proceeds to the `IOException` handler, where the exception is available through `ex` for logging, recovery, or user-facing error handling. This arrangement follows the usual practice of placing more specific exception handlers before handlers for their broader base types, ensuring that applicable I/O failures are handled at the intended level of specificity. Microsoft's API documentation identifies `IOException` as the base class of `DirectoryNotFoundException`: DirectoryNotFoundException Class. The C# language guidance on exception handling also describes how the runtime searches the associated catch clauses for a handler matching the thrown exception type: Exception handling and exceptions.

QUESTION NO: 35

You work as a Software Developer for ABC Inc. The company uses Visual Studio .NET 2008 as its application development platform. You are creating an ASP.NET Web application using .NET Framework 3.5. The application will be accessed by Internet users. You need to enable users to authenticate from the client-side script. You add the following code fragment in the Web.config file of the application:

You must configure the application to ensure that user credentials are validated against

Active Directory by using the client-side script. What will you do to accomplish this?

Each correct answer represents a part of the solution. Choose two.

A. Add the following code fragment to the Web.config file of the application:

B. Configure the application to use the ActiveDirectoryMembershipProvider class.

C. Add the following code fragment to the Web.config file of the application:

D. Configure the application to use the SqlMembershipProvider class.

E. Add the following code fragment to the Web.config file of the application:

ANSWER: B C

Explanation:

Configuring the application to use the ActiveDirectoryMembershipProvider class supplies the membership-validation component that connects ASP.NET membership operations to Active Directory. The provider is designed to validate supplied user names and passwords against an AD domain or AD LDS store, subject to the provider's configured connection, application name, and search settings. This lets the application retain centralized directory credentials rather than maintaining a separate application-specific credential database. Microsoft documents the provider and its configuration capabilities in the ActiveDirectoryMembershipProvider API reference.

Adding `<authentication mode="Forms"></authentication>` is also required because the ASP.NET AJAX AuthenticationService exposes client-side login functionality for Forms Authentication. When the client script invokes the authentication service, ASP.NET uses the configured membership provider to validate the credentials, then establishes the Forms Authentication session for a successful login. This separation is important: Forms Authentication controls the application's web authentication mechanism and ticket issuance, while ActiveDirectoryMembershipProvider performs the backing credential validation against Active Directory. The available ASP.NET authentication modes, including Forms, are described in Microsoft's AuthenticationSection.Mode documentation.

QUESTION NO: 36

Roger works as a Software Developer for BlackWell Inc. He develops an application, named MyApplication1, using Visual Studio .NET. The application contains a form, named MyForm1, which will be used to set a password for a user. A password allotted to a user will be case-sensitive. MyForm1 contains several controls including two TextBox controls, named txtPassword and txtConfirm, respectively. The txtMyPassword control is used to accept a string from a user. The string entered in txtPasword will be used as the password for the user. In txtConfirm, the user has to retype the string that he enters in txtPassword. This will ensure that there are no typing errors. MyForm1 also contains a Button control, named cmdSetPassword, which when clicked will set the password for the user.

Before setting the password for a user, Roger wants to ensure that the two text boxes contain the same string. In order to accomplish this, he decides to use the StrComp function. The password for a user can only be set if the StrComp function returns 0. He has not used an Option Compare statement in the code. Which of the following statements can

Roger use to compare the two strings?

Each correct answer represents a complete solution. Choose two.

A. StrComp(txtPassword.Text, txtConfirm.Text, CompareMethod.Binary)

B. StrComp(txtPassword.Text, txtConfirm.Text, CompareMethod.Text)

C. StrComp(txtPassword.Text, txtConfirm.Text, CompareMethod.CaseSensitive)

D. StrComp(txtPassword.Text, txtConfirm.Text)

ANSWER: A D

Explanation:

`StrComp(txtPassword.Text, txtConfirm.Text, CompareMethod.Binary)` is correct because the `Binary` comparison method compares strings according to the binary values of their characters. Consequently, uppercase and lowercase characters have different values, so strings such as `Password` and `password` do not compare as equal. When both textbox values are identical, `StrComp` returns `0`, which is the required condition before setting a case-sensitive password.

`StrComp(txtPassword.Text, txtConfirm.Text)` is also correct. The optional comparison argument defaults to the comparison behavior specified by `Option Compare`. Because no `Option Compare` statement is present, Visual Basic uses its default `Binary` setting. This makes the omitted-argument call case-sensitive as well, and it returns `0` only when the entered password and confirmation have the same character sequence with the same capitalization. Microsoft documents that `StrComp` returns zero for equal strings and that the default comparison depends on `Option Compare`; Microsoft also documents Binary as the default option when no directive is supplied. See Microsoft.VisualBasic.Strings.StrComp documentation and Option Compare statement documentation.

QUESTION NO: 37

You work as a Software Developer for ABC Inc. You create a Console application named

ConsoleApplication4. You use the System.Security.Cryptography namespace. You want to use the key lengths of 384 bits to 16384 bits. You use RSACryptoServiceProvider class to encrypt and decrypt data. Which of the following code segments will you use to accomplish this task?

Each correct answer represents a part of the solution. Choose all that apply.

A. UnicodeEncoding ByteConverter = new UnicodeEncoding(); byte[] DataToEncrypt = ByteConverter.GetBytes("Encrypt this line for testing"); byte[] encryptData; byte[] decryptData;
RSACryptoServiceProvider RSA = new RSACryptoServiceProvider(); encryptData = RSA.Encrypt(DataToEncrypt, true);
Console.WriteLine("Encrypted : {0}", ByteConverter.GetString(encryptData)); decryptData = RSA.Decrypt(encryptData,false);
Console.WriteLine("Decrypted : {0}", ByteConverter.GetString(decryptData));

B. UnicodeEncoding ByteConverter = new UnicodeEncoding(); byte[] DataToEncrypt = ByteConverter.GetBytes("Encrypt this line for testing"); byte[] encryptData; byte[] decryptData;

RSACryptoServiceProvider RSA = new RSACryptoServiceProvider(); encryptData = RSA.Encrypt(DataToEncrypt, false); Console.WriteLine("Encrypted : {0}", ByteConverter.GetString(encryptData)); decryptData = RSA.Decrypt(encryptData,false); Console.WriteLine("Decrypted : {0}", ByteConverter.GetString(decryptData));

C. UnicodeEncoding ByteConverter = new UnicodeEncoding(); byte[] DataToEncrypt = ByteConverter.GetBytes("Encrypt this line for testing"); byte[] encryptData; byte[] decryptData;
RSACryptoServiceProvider RSA = new RSACryptoServiceProvider(); encryptData = RSA.Encrypt(DataToEncrypt, false); Console.WriteLine("Encrypted : {0}", ByteConverter.GetString(encryptData)); decryptData = RSA.Decrypt(encryptData,true); Console.WriteLine("Decrypted : {0}", ByteConverter.GetString(decryptData));

D. UnicodeEncoding ByteConverter = new UnicodeEncoding(); byte[] DataToEncrypt = ByteConverter.GetBytes("Encrypt this line for testing"); byte[] encryptData; byte[] decryptData;
RSACryptoServiceProvider RSA = new RSACryptoServiceProvider(); encryptData = RSA.Encrypt(DataToEncrypt, true); Console.WriteLine("Encrypted : {0}", ByteConverter.GetString(encryptData)); decryptData = RSA.Decrypt(encryptData,true); Console.WriteLine("Decrypted : {0}", ByteConverter.GetString(decryptData));

ANSWER: B D

Explanation:

The code segments that use `RSA.Encrypt(DataToEncrypt, false)` together with `RSA.Decrypt(encryptData, false)`, and those that use `RSA.Encrypt(DataToEncrypt, true)` together with `RSA.Decrypt(encryptData, true)`, are correct because the Boolean argument selects the RSA padding scheme and must be consistent for the encryption/decryption pair. Passing `false` uses PKCS #1 v1.5 encryption padding. Passing `true` requests OAEP padding (where supported by the underlying cryptographic service provider). In each valid segment, ciphertext is produced with one selected padding format and then decoded using that same format, allowing the private key operation to recover the original plaintext bytes. The resulting byte array can then be converted back to text through the same `UnicodeEncoding` instance. `RSACryptoServiceProvider` supports RSA key sizes within the provider's supported legal key-size range; the referenced range encompasses the stated 384-bit through 16384-bit requirement, subject to the installed provider and platform capabilities. Microsoft documents the Boolean padding parameter for both methods and specifies that the same padding choice is required for compatible encryption and decryption operations. See RSACryptoServiceProvider.Encrypt and RSACryptoServiceProvider.Decrypt.

QUESTION NO: 38

You work as a Software Developer for Webtech Inc. You plan to migrate the existing Active Server Pages (ASP) pages to ASP.NET Web pages in order to implement rich functionalities of a Web browser in your Web pages. Therefore, you want to add controls to Web pages of your existing application named WebApp1. Which of the following controls will be most suitable to implement?

A. HTML server controls

B. Web server controls

C. Validation controls

D. HTML controls

ANSWER: B

Explanation:

Web server controls are the most suitable choice because they provide ASP.NET Web Forms with a rich, event-driven programming model while rendering appropriate HTML for the client browser. These controls are declared with the `runat="server"` attribute and are processed on the server, allowing the application to use strongly typed properties, methods, server-side events, state management, and reusable control behavior. This model is particularly useful when modernizing classic ASP pages because it separates server-side application logic from the browser markup and lets developers build interactive pages using familiar object-oriented C# code. For example, controls such as buttons, text boxes, lists, calendars, and data-bound controls expose a consistent server-side API and can participate in the ASP.NET page life cycle. Their rendering can also adapt to browser capabilities, supporting the goal of providing richer browser-facing functionality without requiring the developer to manually generate every HTML element. Microsoft's overview of ASP.NET

Web Forms describes server controls as the core mechanism for creating interactive, event-based web applications: ASP.NET Web Server Controls Overview. The System.Web.UI.WebControls namespace reference documents the standard ASP.NET web server control classes available for this purpose.

QUESTION NO: 39

You work as a Web Application Developer for ABC Inc. The company uses Visual Studio

.NET as its application development platform. You create a Web application using .NET Framework 2.0. The application is for members only. The behavior of the Web application varies according to the role of the user. The Web application uses the ASP.NET Membership control for creation of user accounts. You are required to find out whether a user is a member of a specific role. What will you do?

A. Pass the role names to the User.IsInRole method.

B. Pass the user names and passwords to the Membership.ValidateUser method.

C. Pass the role names to the Roles.RoleExists method.

D. Pass the user names to the User.IsUserInRole method.

ANSWER: A

Explanation:

Pass the role names to the User.IsInRole method. is correct because ASP.NET exposes the currently authenticated request identity through the `User` principal. Its `IsInRole` method determines whether that current user belongs to the specified role, allowing application code to select role-dependent behavior or authorize access to functionality. For example, code can use `if (User.IsInRole("Administrators"))` before presenting or performing an administrative operation. This approach uses the role information attached to the authenticated principal by ASP.NET role management, rather than attempting to authenticate the user again. Membership and roles serve related but distinct purposes: membership establishes and manages user accounts, while role management associates users with named groups used for authorization decisions. The role provider must be configured and role management enabled so that ASP.NET can populate the principal's role information during the request. Microsoft documents `IPrincipal.IsInRole` as the mechanism for determining whether the current principal belongs to a specified role, and ASP.NET role management as the feature for managing role-based authorization. See Microsoft's IPrincipal.IsInRole documentation and ASP.NET role management documentation.

QUESTION NO: 40

You work as a Software Developer for ABC Inc. You develop a Windows application named

MyApp1. MyApp1 contains a Windows form named MyWinForm1. The form uses several controls such as Button, TextBox, and Label. You want mouse click events to be raised in the form. You want to use the mouse events that pass an instance of EventArgs in the signature of the event handler of a Button control. Which of the following events will you use in the application to accomplish the task?

Each correct answer represents a complete solution. Choose three.

A. MouseMove

B. MouseDown

C. MouseHover

D. MouseEnter

E. MouseUp

F. MouseLeave

ANSWER: C D F

Explanation:

`MouseHover`, `MouseEnter`, and `MouseLeave` are the Button mouse-related events whose handlers use the standard `EventHandler` delegate signature: `void Handler(object sender, EventArgs e)`. This makes them appropriate where the requirement specifically calls for an `EventArgs` instance rather than mouse-coordinate and button-state information. `MouseEnter` is raised when the pointer enters the control's client area, `MouseLeave` is raised when it exits, and `MouseHover` is raised when the pointer pauses over the control. Although these events are mouse interactions, they do not need to expose cursor position, pressed buttons, click counts, or wheel data; consequently, their event data is represented by the base `EventArgs` type. A handler can be attached in code, for example, with `button.MouseEnter += Handler;`, provided that `Handler` has the required `object`/`EventArgs` signature. The .NET Button documentation lists these events and their associated inherited Control event model, while the EventHandler documentation defines the standard delegate pattern used for events without specialized event data. See Control.MouseEnter and System.EventHandler.

QUESTION NO: 41

You work as a Software Developer for ABC Inc. The company uses Visual Studio.NET

2005 as its application development platform. You have recently finished development of Web services using the .NET Framework. The access control to Web services is part of your accountability. You are currently revealing an existing class as a XML Web service. You are required to ensure that the XML Web service is exclusively accessible to Web service clients within the ABC Inc. domain. To fulfill this requirement you want to change the access modifiers on methods that must be represented as Web methods. What will you do to accomplish this task?

A. Use the Public access modifier for each Web method.

B. Use the Private access modifier for each Web method.

C. Use the Internal or Friend access modifier for each Web method.

D. Use the Protected access modifier for each Web method.

ANSWER: A

Explanation:

Use the Public access modifier for each Web method. An ASP.NET XML Web Service exposes operations by applying `WebMethodAttribute` to methods that are publicly callable. Public visibility is therefore the required method-level access modifier for an operation to be published as a Web service method and invoked by a remote Web service client. The .NET Framework documentation for `WebMethodAttribute` describes its purpose as marking methods as callable through XML Web services; the service infrastructure must be able to invoke the marked method outside the declaring class. See Microsoft's WebMethodAttribute documentation.

Public method visibility does not, by itself, grant unrestricted use of the service. The requirement to limit clients to the ABC Inc. domain is enforced separately at the Web application or IIS layer, commonly through Windows Authentication and authorization rules that allow the appropriate domain users or groups. IIS authentication determines the caller identity before the service operation is used, while the public modifier enables the service framework to expose the operation correctly. Microsoft documents this authentication model in its IIS Windows Authentication reference.