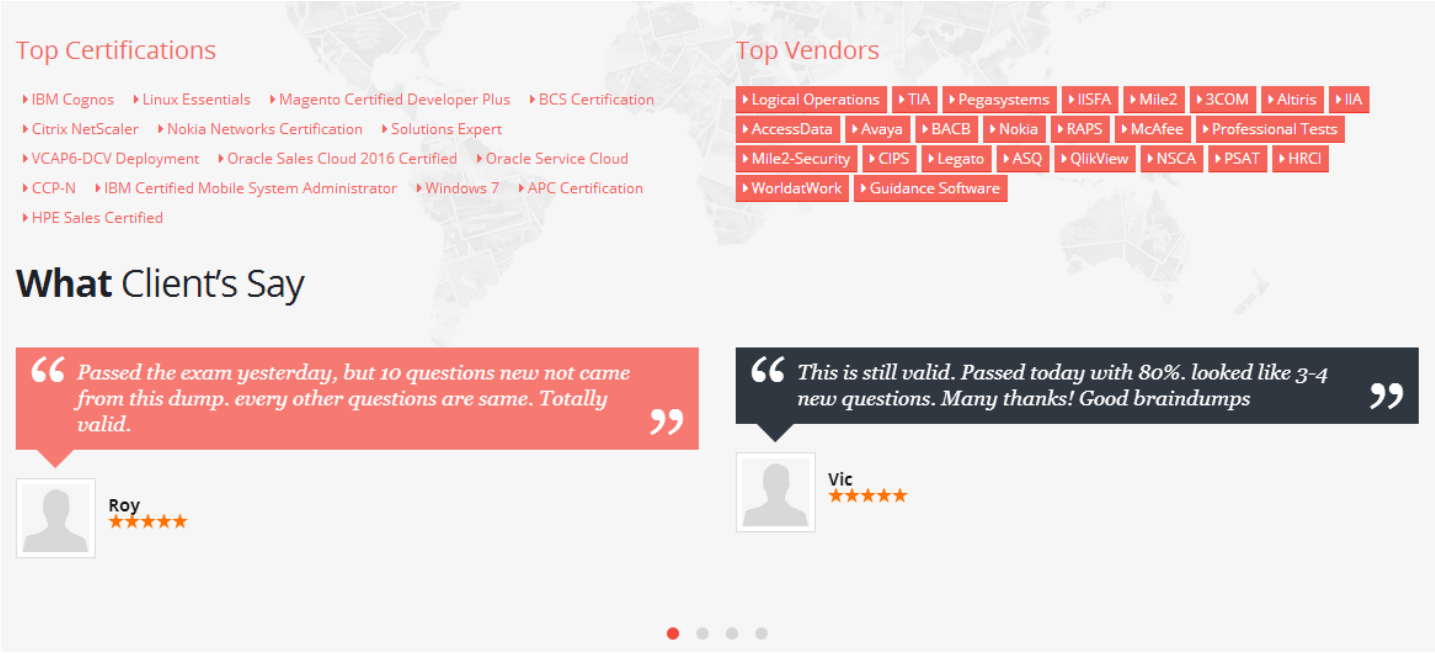
Lead2passExam



http://www.lead2passexam.com/

Available Exam Cram and Valid Dumps - Lead2Pass Exam

Exam : **70-483J**

Title : Programming in C# (70-483日本語版)

Vendor : Microsoft

Version : DEMO

QUESTION NO: 1
アプリケーションは、次の形式でJSONデータを受信します。

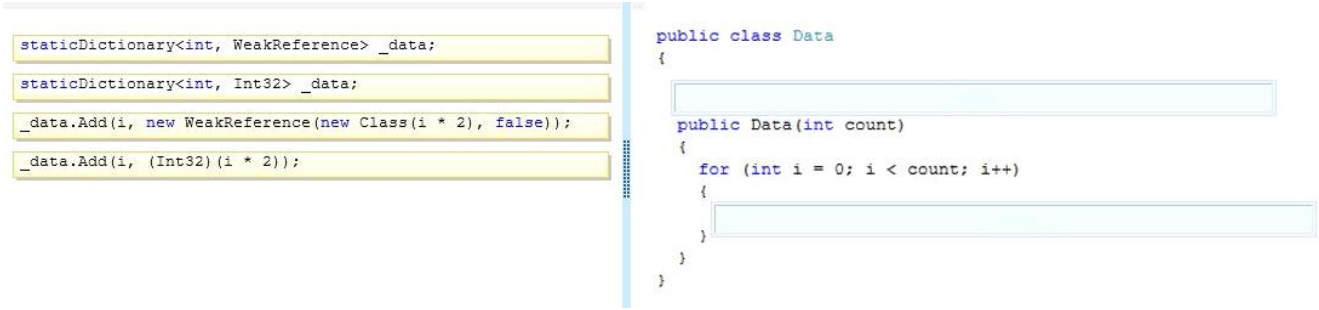
```
{ "FirstName" : "David",
  "LastName" : "Jones",
  "Values" : [0, 1, 2] }
```

アプリケーションは、次のコードセグメントを含みます。（行番号は参考のために含まれるだけです。）

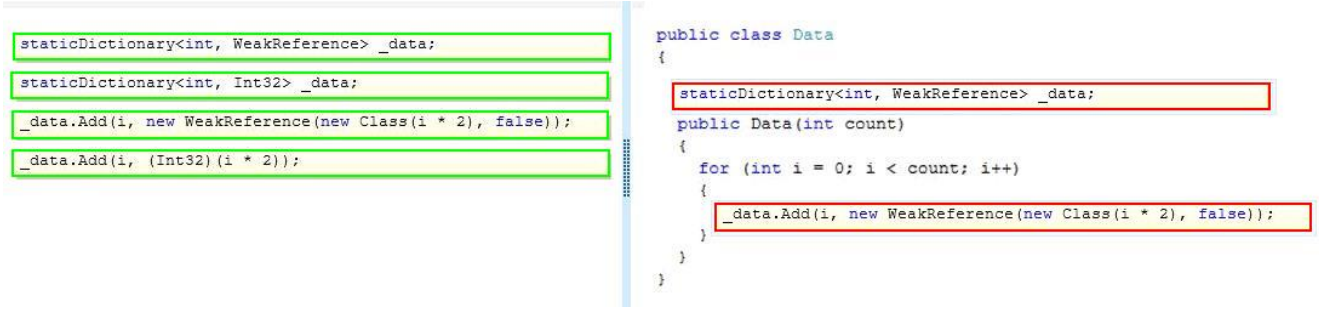
```
01 public class Name
02 {
03    public int[] Values { get; set; }
04    public string FirstName { get; set; }
05    public string LastName { get; set; }
06 }
07 public static Name ConvertToName(string json)
08 {
09    var ser = new JavaScriptSerializer();
10
11 }
```

あなたはConvertToName（）メソッドは、名前のオブジェクトとしてJSON入力文字列を返すことを確認する必要があります。
あなたは、第10行でどのコード部分を挿入するべきですか？
A. Return ser.ConvertToType<Name>(json);
B. Return ser.DeserializeObject(json);
C. Return ser.Deserialize<Name>(json);
D. Return (Name)ser.Serialize(json);
Answer: C
Explanation:
JavaScriptSerializer.Deserialize<T> - Converts the specified JSON string to an object of type T.
http://msdn.microsoft.com/en-us/library/bb355316.aspx

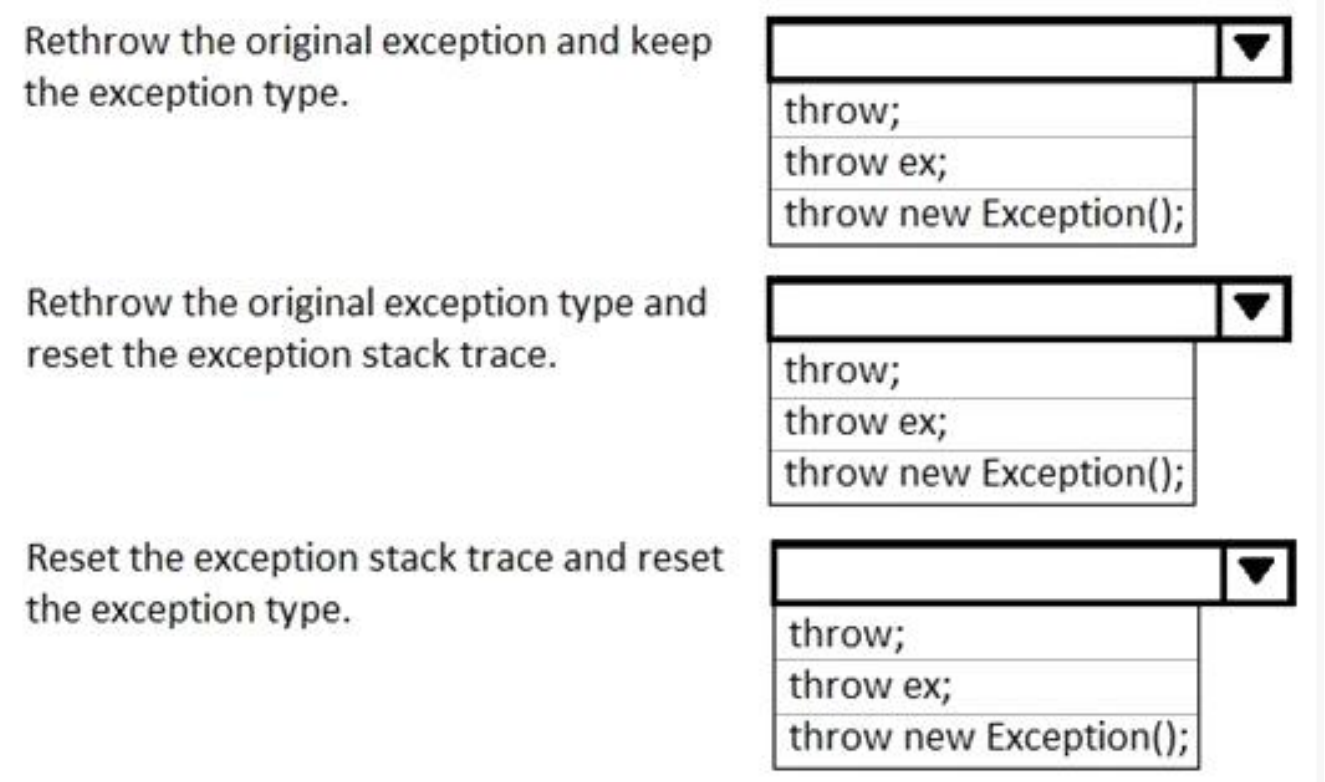
QUESTION NO: 2
_dataという名前の辞書オブジェクトを含むDataという名前のクラスを作成しています。
ガベージコレクションプロセスで_dataオブジェクトの参照を収集できるようにする必要があります。
関連コードをどのように完成させるべきですか？
（回答するには、適切なコードセグメントを回答エリアの正しい場所にドラッグします。各コードセグメントは、1回、複数回、またはまったく使用されないことがあります。）



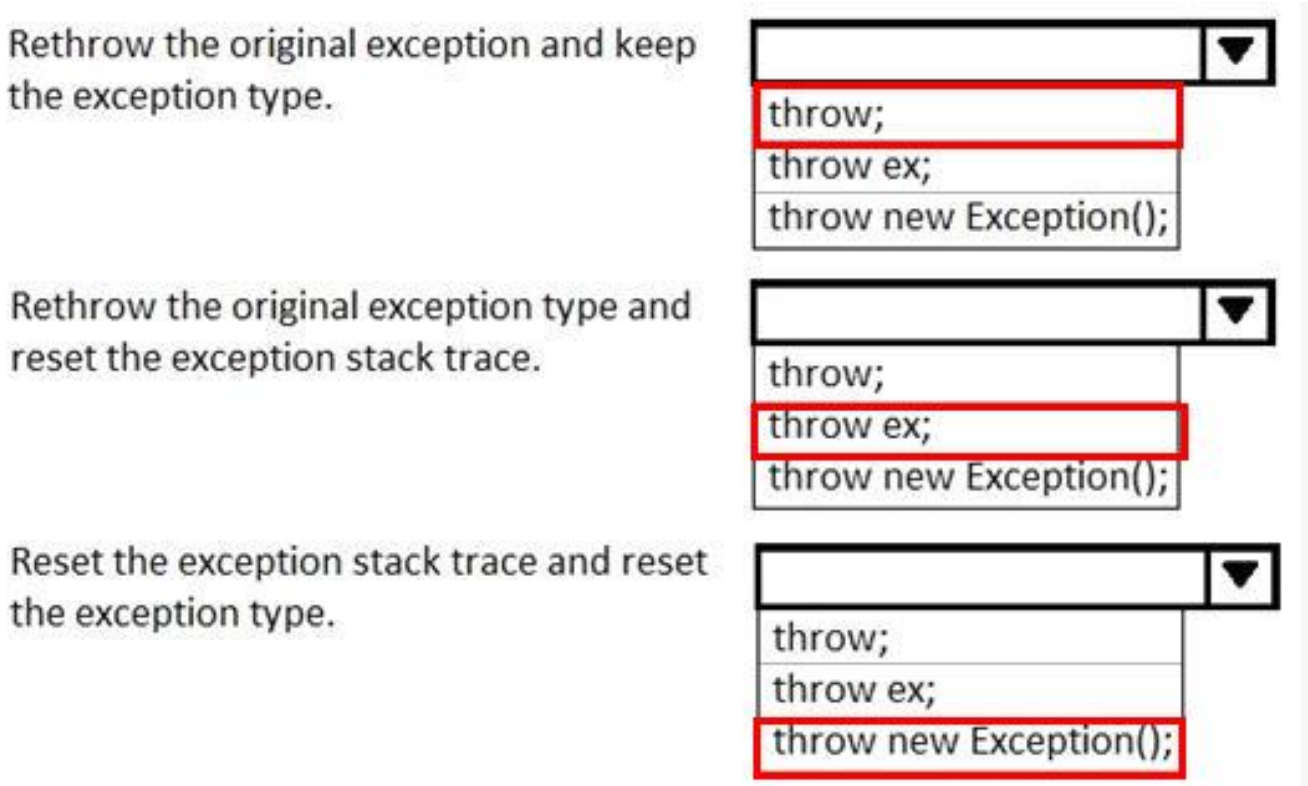
Answer:



QUESTION NO: 3
C＃アプリケーションの例外を処理するコードを記述しています。
exという名前の例外を処理するさまざまな方法を特定する必要があります。
各作業にどのコード行を使用する必要がありますか？
答えるには、回答エリアの各タスクに適切なコード行を選択します。



Answer:



Explanation:
References: https://blogs.msdn.microsoft.com/perfworld/2009/06/15/how-can-i-throw-an-exception-without-losing-the-original-stack-trace-information-in-net/

QUESTION NO: 4
あなたはアプリケーションを開発しています。
このアプリケーションには、EmployeeおよびPersonという名前のクラスと、IPersonという名前のインタフェースが含まれています。
Employeeクラスは次の要件を満たしている必要があります:
*
Personクラスから継承するか、またはIPersonインターフェースを実装する必要があります。
* アプリケーション内の他のクラスによって継承可能でなければなりません。
Employeeクラスが要件を満たしていることを確認する必要があります。
この目標を達成するためにどの2つのコードセグメントを使用できますか？
（それぞれの正解は完全な解答を提示しますが、2つを選択してください）。

A.
```
sealed class Employee : Person
{
   ...
}
```

B.
```
abstract class Employee : Person
{
   ...
}
```

C.
```
sealed class Employee : IPerson
{
   ...
}
```

D.
```
abstract class Employee : IPerson
{
   ...
}
```

A. Option A
B. Option B
C. Option C
D. Option D
Answer: B,D
Explanation:
Sealed - When applied to a class, the sealed modifier prevents other classes from inheriting from it.
Reference:
http://msdn.microsoft.com/en-us/library/88c54tsw(v=vs.110).aspx

QUESTION NO: 5
あなたは、GetValidEmailAddressesという名前をつけられる方法を実装しています。GetValidEmailAddresses ( ) 方法は、電子メールaddressesを代表する文字列値のリストを処理します。
GetValidEmailAddresses ( ) 方法は、有効なフォーマットである電子メール・アドレスだけを返さなければなりません。
あなたは、GetValidEmailAddresses ( ) 方法を実装する必要があります。

あなたは、このゴールを達成するために、どの二つコード部分を使うことができますか？ (各正解は完全なソリューションを提供します。2を選択してください。)

A.
```
private static List<String> GetValidEmailAddresses(string input, string pattern)
{
  var regex = new Regex(pattern);
  var matches = regex.Matches(input);
  var validEmailAddresses = new List<String>();
  foreach(Match match in matches)
  {
    if(!match.Success)
    {
      validEmailAddresses.Add(match.Value);
    }
  }
  return validEmailAddresses;
}
```

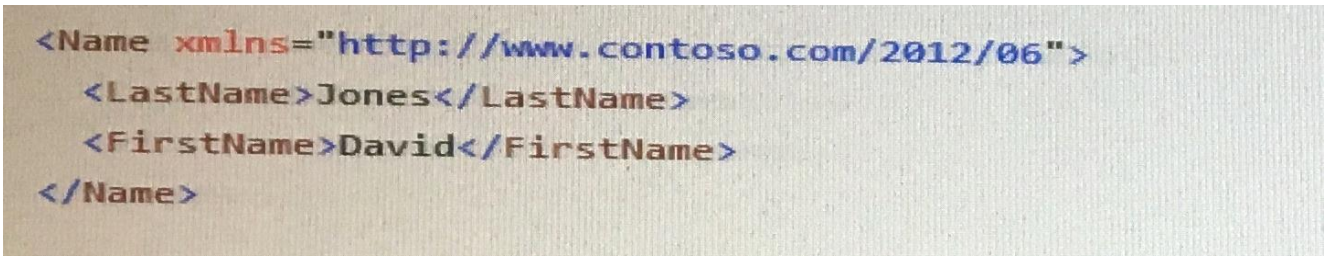
B.
```
private static List<String> GetValidEmailAddresses(string input, string pattern)
{
  var regex = new Regex(pattern);
  var matches = regex.Matches(input);
  return (from Match match in matches where match.Success select match.Value).ToList();
}
```

C.
```
private static List<String> GetValidEmailAddresses(string input, string pattern)
{
  var regex = new Regex(pattern);
  var matches = regex.Matches(input);
  return (from Match match in matches where match.Success select match.Success.ToString()).ToList();
}
```

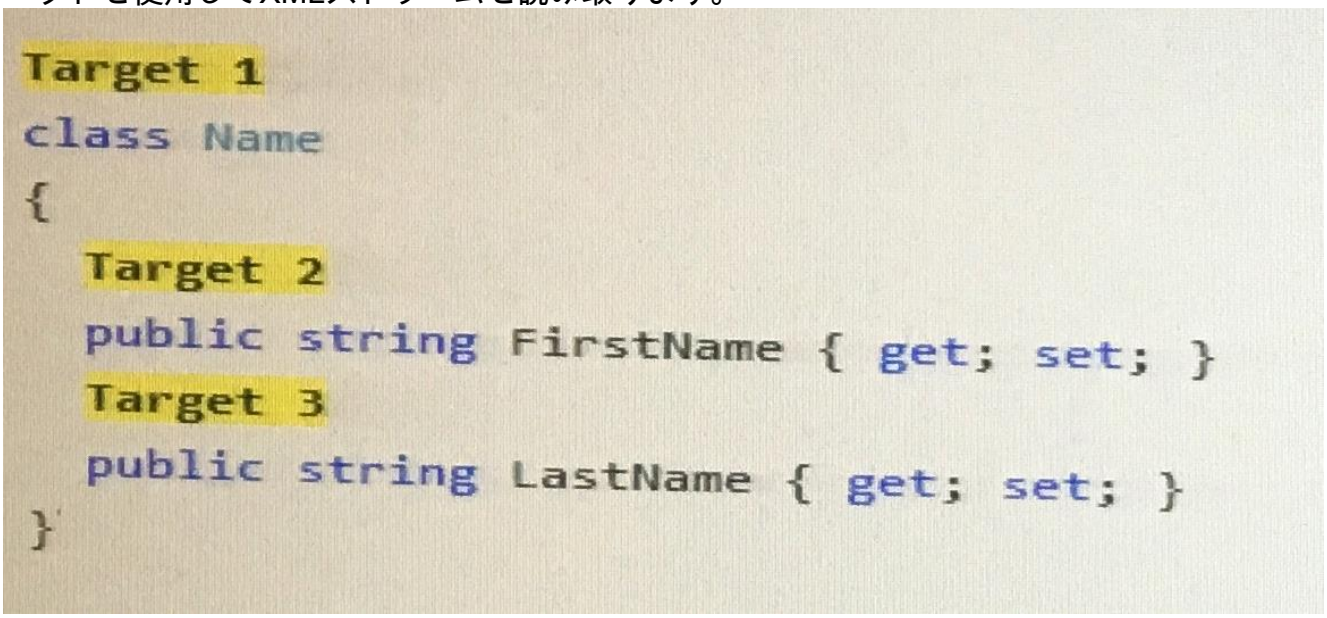
D.
```
private static List<String> GetValidEmailAddresses(string input, string pattern)
{
  var regex = new Regex(pattern);
  var matches = regex.Matches(input);
  var validEmailAddresses = new List<String>();
  foreach(Match match in matches)
  {
    if(match.Success)
    {
      validEmailAddresses.Add(match.Value);
    }
  }
  return validEmailAddresses;
}
```

A. Option A
B. Option B
C. Option C
D. Option D
Answer: B,D
Explanation:
Note:
* List<T>.Add Method
Adds an object to the end of the List<T>.

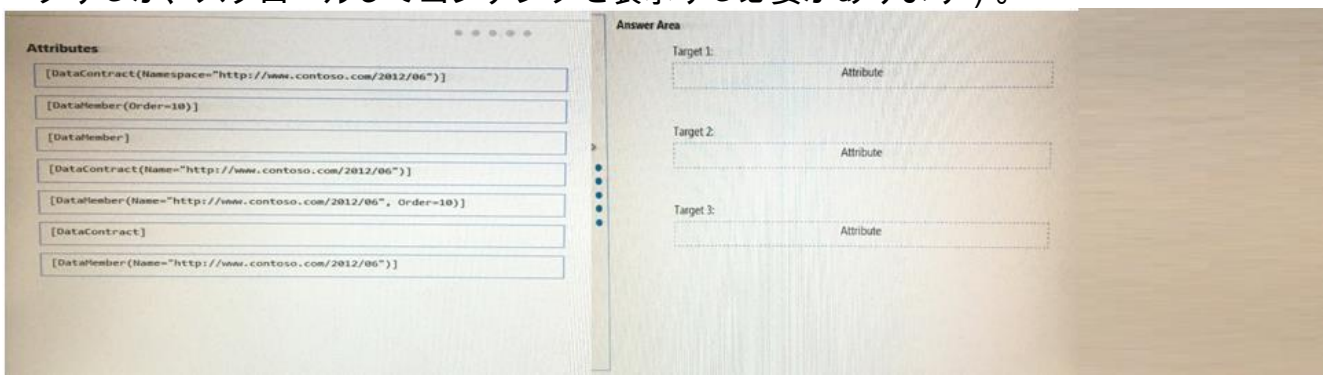
QUESTION NO: 6
アプリケーションは、ストリームからXMLを直列化および逆直列化します。
XMLストリームの形式は次のとおりです。



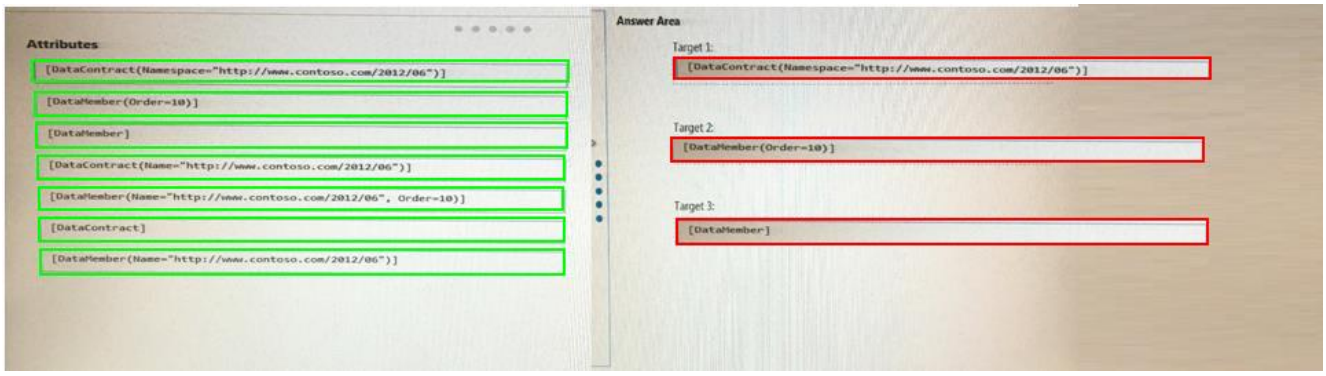
アプリケーションは、次のコードセグメントで宣言されているDataContractSeria.izerオブジェクトを使用してXMLストリームを読み取ります。



コードを完成させるために、ターゲット1とターゲット2にどの属性を含めるべきですか？（回答するには、適切な属性を回答エリアの正しいターゲットにドラッグします。各属性は1回、複数回、またはまったく使用されないことがあります。分割バーをペインの間にドラッグするか、スクロールしてコンテンツを表示する必要があります）。



Answer:



QUESTION NO: 7
あなたは、いくつかのアプリケーションにより用いられるAccountという名前をつけられるクラスを開発しています。
Accountクラスを消費するアプリケーションは、いくつかの異なる方法を実行するというAccountクラスへの非同期要求をします。
あなたは、方法への1つの呼び出しだけが一度に実行されることを確実とする必要があります。
あなたは、どのキーワードを使用しなければなりませんか？
A. sealed
B. protected
C. checked
D. lock
Answer: D
Explanation:
The lock keyword ensures that one thread does not enter a critical section of code while another thread is in the critical section. If another thread tries to enter a locked code, it will wait, block, until the object is released.
Reference:
https://msdn.microsoft.com/en-us/library/c5kehkcz.aspx

QUESTION NO: 8
LoanクラスというクラスのLoanCollectionという名前のカスタムコレクションを開発しています。
foreachループを使用して、LoanCollectionコレクション内の各Loanオブジェクトを処理できることを確認する必要があります。
関連コードをどのように完成させるべきですか？
（回答するには、適切なコードセグメントを回答エリアの正しい場所にドラッグします。各コードセグメントは、1回、複数回、またはまったく使用されないことがあります。）

```
: IComparable
: IEnumerable
: IDisposable
public IEnumerator GetEnumerator()
public int CompareTo(object obj)
public void Dispose()
_loanCollection[0].Amount++;
return obj == null ? 1 : _loanCollection.Length;
return _loanCollection.GetEnumerator();
```

```
public class LoanCollection
{
    private readonly Loan[] _loanCollection;
    public LoanCollection(Loan[] loanArray)
    {
        _loanCollection = new Loan[loanArray.Length];

        for (int i = 0; i < loanArray.Length; i++)
        {
            _loanCollection[i] = loanArray[i];
        }
    }

    {

    }
}
```

Answer:

```
: IComparable
: IEnumerable
: IDisposable
public IEnumerator GetEnumerator()
public int CompareTo(object obj)
public void Dispose()
_loanCollection[0].Amount++;
return obj == null ? 1 : _loanCollection.Length;
return _loanCollection.GetEnumerator();
```

```
public class LoanCollection : IEnumerable
{
    private readonly Loan[] _loanCollection;
    public LoanCollection(Loan[] loanArray)
    {
        _loanCollection = new Loan[loanArray.Length];

        for (int i = 0; i < loanArray.Length; i++)
        {
            _loanCollection[i] = loanArray[i];
        }
    }

    public IEnumerator GetEnumerator()
    {
        return _loanCollection.GetEnumerator();
    }
}
```

QUESTION NO: 9
あなたは、いくつかの物を使うアプリケーションを開発しています。
アプリケーションは、以下のコード部分を含みます。（行番号は参考のために含まれるだけです。）

```
01 private bool IsNull(object obj)
02 {
03
04    return false;
05 }
```

あなたは、物が無効かどうかについて評価する必要があります。
あなたは、線03にどのコード部分を挿入しなければなりませんか？

A.
```
if (obj = null)
{
  return true;
}
```

B.
```
if (null)
{
  return true;
}
```

C.
```
if (obj ==  0)
{
  return true;
}
```

D.
```
if (obj == null)
{
  return true;
}
```

A. Option A
B. Option B
C. Option C
D. Option D
Answer: D
Explanation:
Use the == operator to compare values and in this case also use the null literal.

QUESTION NO: 10
あなたは、長時間実行されているデータの処理動作を起動するTask.Run（ ）方法を使用します。データ処理操作は、多くの場合、大量のネットワーク輻輳時に失敗します。
データ処理操作が失敗した場合、第2の動作は、最初の操作のいずれかの結果をクリーンアップする必要があります。
あなたは、データ処理活動がならされてない例外を投げる場合だけ、2回目の活動が実施されることを確実とする必要があります。
あなたは、何をするべきですか?
A. Create a TaskCompletionSource<T> object and call the TrySetException() method of the object.
B. Create a task by calling the Task.ContinueWith() method.
C. Examine the Task.Status property immediately after the call to the Task.Run() method.

D. Create a task inside the existing Task.Run() method by using the AttachedToParent option.

Answer: B

Explanation:

Task.ContinueWith - Creates a continuation that executes asynchronously when the target Task completes.The returned Task will not be scheduled for execution until the current task has completed, whether it completes due to running to completion successfully, faulting due to an unhandled exception, or exiting out early due to being canceled.

http://msdn.microsoft.com/en-us/library/dd270696.aspx

QUESTION NO: 11

注：この質問は、同じシナリオを提示する一連の質問の一部です。

シリーズの各質問には、記載された目標を達成できる独自の解決策が含まれています。

いくつかの質問セットには1つ以上の正しい解決策があるかもしれないが、他の質問セットには正しい解決策がないかもしれない。

このセクションの質問に答えると、それに戻ることはできません。

その結果、これらの質問はレビュー画面に表示されません。

あなたは次のC＃コードを持っています。（行番号は参照用にのみ記載されています）。

```
01    int[] intArray = { 1, 2, 3, 4, 5 };
02
03    foreach (var item in intArray)
04    {
05        Console.WriteLine(item);
06    }
```

次の出力に示すように、配列要素の合計を表示するにはforeachループが必要です。

1
3
6
10
15

解決策：02行目に次のコードを挿入します。

```
for (int i = 1; i < intArray.Length; i++)
{
 intArray[i] += intArray[i-1];
}
```

これは目標を満たしていますか？

A. Yes

B. No

Answer: B

QUESTION NO: 12
値をコレクションに格納する必要があります。
ソリューションは次の要件を満たしている必要があります:
* 値は、コレクションに追加された順序で格納する必要があります。
* 値は、先入れ先出しの順序でアクセスする必要があります。
あなたはコレクションのどのタイプを使うべきであるか？
A. SortedList
B. Queue
C. ArrayList
D. Hashtable
Answer: B
Explanation:
The Queue class implements a queue as a circular array. Objects stored in a Queue are inserted at one end and removed from the other.
Queues and stacks are useful when you need temporary storage for information; that is, when you might want to discard an element after retrieving its value. Use Queue if you need to access the information in the same order that it is stored in the collection.

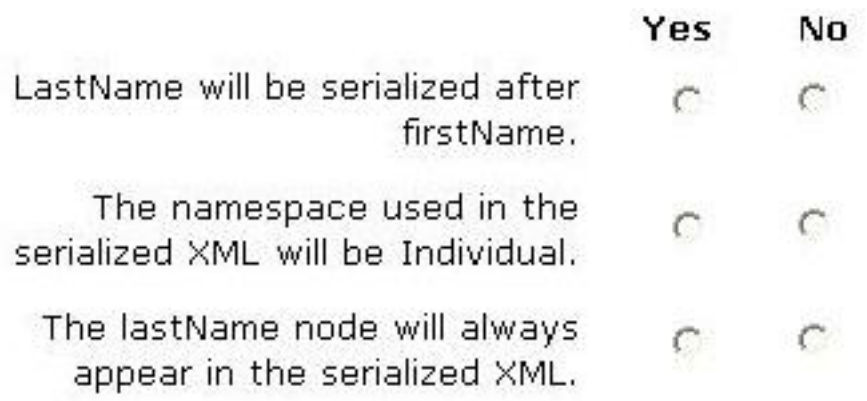
QUESTION NO: 13
ProcessDataという名前の新しいメソッドを実装しています。
ProcessData（）メソッドは、Webサービスから在庫情報を取得するために長時間実行される第三者コンポーネントを呼び出します。
サードパーティのコンポーネントはIAsyncRcsultパターンを使用して、長時間実行される操作の完了を通知し、UIを新しい値で更新できるようにします。
あなたは、UIスレッドを塞ぐことを避けるために呼び出しコードが長期の操作を5ystem.Threading_Tasks.Taskオブジェクトとしてを処理することを保証する必要がある。
あなたはどの2回の行動を実行するべきであるか？（各正解は解の一部を表しています。2つを選択してください。）
A. async修飾子をProcessData（）メソッドのシグネチャに適用します。
B. TaskFactory FromAsync（）メソッドを使用してコンポーネントを呼び出します。
C. ProcessDataOメソッドのシグネチャに次の属性を適用します:
[MethodImpl (MetrhodImplOptiions . Synchronized)]
D. TaskCompletionSource <T>オブジェクトを作成します。
Answer: B,D

QUESTION NO: 14
次のコード行があります。 Type type1 = typeof^MyClass);
タイプ1のタイプを持つobjという名前のオブジェクトを作成する必要があります。
あなたはどのコード行を使うべきであるか？
A. object obj = Activator .CreateInstance("type 1" .GetType()) ;
B. type1 obj = Activator.CreateInstance(typel);
C. type1 obj = Activator.CreateInstance("type1" .GetType());
D. object obj = Activator.CreateInstance(typ1);
Answer: B

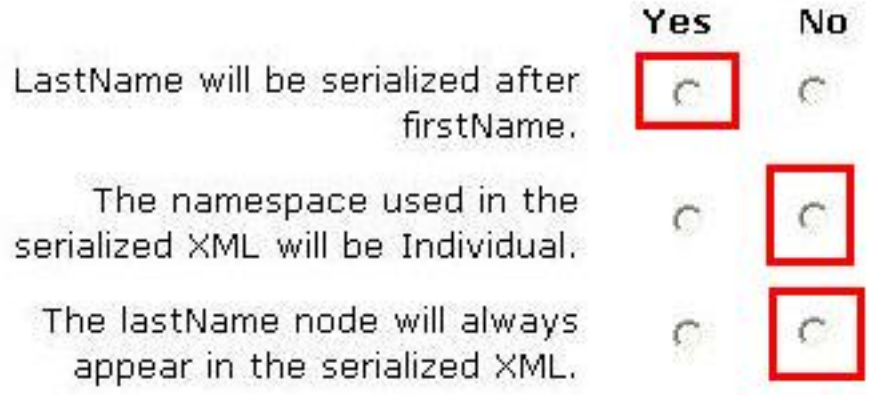
QUESTION NO: 15
あなたは次のコードを持っています：

```
[DataContract(Name="Indvidual")]
  public class Individual
  {
    private string m_FirstName;
    private string m_LastName;

    [DataMember]
    public string FirstName
    {
      get { return m_FirstName; }
      set { m_FirstName = value; }
    }

    [DataMember(EmitDefaultValue=false)]
    public string LastName
    {
      get { return m_LastName; }
      set { m_LastName = value; }
    }


    public Individual()
    {
    }

   public Individual(string firstName, string lastName)
    {
      m_FirstName = firstName;
      m_LastName = lastName;
    }

  }
```

次の各文について、その文が真であればYesを選択します。 それ以外の場合は、「いいえ」



Answer:



Explanation:
Note:
* The System.Runtime.Serialization namespace contains classes that can be used for serializing and deserializing objects. Serialization is the process of converting an object or a graph of objects into a linear sequence of bytes for either storage or transmission to another location. Deserialization is the process of taking in stored information and recreating objects from it.
* EmitDefaultValue
DataMemberAttribute.EmitDefaultValue Property
Gets or sets a value that specifies whether to serialize the default value for a field or property being serialized.
true if the default value for a member should be generated in the serialization stream; otherwise, false.

QUESTION NO: 16
Microsoft .NET
Frameworkアセンブリに関する情報を取得するアプリケーション用のコードを開発しています。
次のコードセグメントはアプリケーションの一部です（行番号は参照用にのみ含まれています）。

```
01 public void ViewMetadata(string filePath)
02 {
03   var bytes = File.ReadAllBytes(filePath);
04
05   ...
06 }
```

04行目にコードを挿入する必要があります。
コードはアセンブリをロードする必要があります。
アセンブリがロードされると、コードはアセンブリメタデータを読み取ることができなければなりませんが、コードはアセンブリから実行コードへのアクセスを拒否されなければなりません。
あなたはどのコードセグメントをライン04に挿入するべきであるか？

A. Assembly.ReflectionOnlyLoadFrom(bytes);

B. Assembly.ReflectionOnlyLoad(bytes);

C. Assembly.Load(bytes);

D. Assembly.LoadFrom(bytes);

Answer: B

Explanation:
The Assembly.ReflectionOnlyLoad method (Byte[]) loads the assembly from a common object file format (COFF)-based image containing an emitted assembly. The assembly is loaded into the reflection-only context of the caller's application domain.
You cannot execute code from an assembly loaded into the reflection-only context.
Incorrect:
Not A: The Assembly.ReflectionOnlyLoadFrom method (String) loads an assembly into the reflection-only context, given its path.
Reference:
https://msdn.microsoft.com/en-us/library/h55she1h(v=vs.110).aspx

QUESTION NO: 17
あなたは、スコアカードという名前のクラスを開発しています。
次のコードは、スコアカードのクラスを実装します。（線番号は参考のために含まれるだけです。）

```
01 public class Scorecard
02 {
03   private Dictionary<string, int> players = new Dictionary<string, int>();
04   public void Add(string name, int score)
05   {
06     players.Add(name, score);
07   }
08
09 }
```

あなたは、スコアカードのクラスの実装をテストするには、以下のユニット·テスト·メソッドを作成します：

```
[TestMethod]
public void UnitTest1()
{
  Scorecard scorecard = new Scorecard();
  scorecard.Add("Player1", 10);
  scorecard.Add("Player2", 15);
  int expectedScore = 15;
  int actualScore = scorecard["Player2"];
  Assert.AreEqual(expectedScore, actualScore);
}
```

あなたは、単位テストが通ることを確実とする必要があります。
あなたは何をするべきですか？

A. Insert the following code segment at line 08:

```
public int this[string name]
{
  get
  {
    return players[name];
  }
}
```

B. Insert the following code segment at line 08:

```
public Dictionary<string, int> Players
{
  get
  {
    return players;
  }
}
```

C. Replace line 03 with the following code segment:

```
public Dictionary<string, int> Players = new Dictionary<string, int>();
```

D. Insert the following code segment at line 08:

```
public int score(string name)
{
  return players[name];
}
```

A. Option A

B. Option B
C. Option C
D. Option D
Answer: A
Explanation:
You need to add indexer to the class.

QUESTION NO: 18
あなたは、C＃にアプリケーションを開発しています。
アプリケーションは、整数を使用して、数学的計算を実行するために使用される方法に例外処理を使用します。
あなたはこの方法については、次のcatchブロックを書き込みます（線番号は、参考のために含まれるだけです）：

```
01
02 catch(ArithmeticException e) {Console.WriteLine("Arithmetic error");}
03
04 catch(ArgumentException e) {Console.WriteLine("Bad Argument");}
05
06 catch(Exception e) {Console.WriteLine("General error");}
07
```

あなたは、メソッドに次のコードを追加する必要があります：

```
catch(DivideByZeroException e) {Console.WriteLine("Divide by zero");}
```

どの線に、あなたはコードを挿入しなければなりませんか？
A. 01
B. 03
C. 05
D. 07
Answer: A

QUESTION NO: 19
あなたは、C＃を使用してアプリケーションを開発しています。
アプリケーションが長時間実行される処理を行うオブジェクトが含まれています。
あなたは、プロセスが完了するまで、ガベージコレクタがオブジェクトのリソースを解放しないようにする必要があります。
あなたはどのガベージコレクタメソッドを使用するべきですか？
A. WaitForFullGCComplete()
B. SuppressFinalize()
C. collect()
D. RemoveMemoryPressure()
Answer: B
Explanation:
You can use the SuppressFinalize method in a resource class to prevent a redundant garbage collection from being called.
Reference:
https://msdn.microsoft.com/en-us/library/system.gc.suppressfinalize(v=vs.110).aspx

QUESTION NO: 20
あなたはconfigファイルを使用するアプリケーションを開発しています。
次のようにconfigファイルの関連する部分が示されています：

```
<system.diagnostics>
  <trace autoflush="false" indentsize="0">
    <listeners>
      <add name="appListener"
        type="System.Diagnostics.EventLogTraceListener"
        initializeData="TraceListenerLog" />
    </listeners>
  </trace>
</system.diagnostics>
```

アプリケーションが.configファイルに指定されている設定を使用してイベントTOGに書き込むためにその診断データを確認する必要があります。
あなたは、何をアプリケーションコードに含むべきですか？

A.
```
EventLog log = new EventLog();
log.WriteEntry("Trace data...");
```

B.
```
Debug.WriteLine("Trace data...");
```

C.
```
Console.SetOut(new StreamWriter("System.Diagnostics.EventLogTraceListener"));
Console.WriteLine("Trace data...");
```

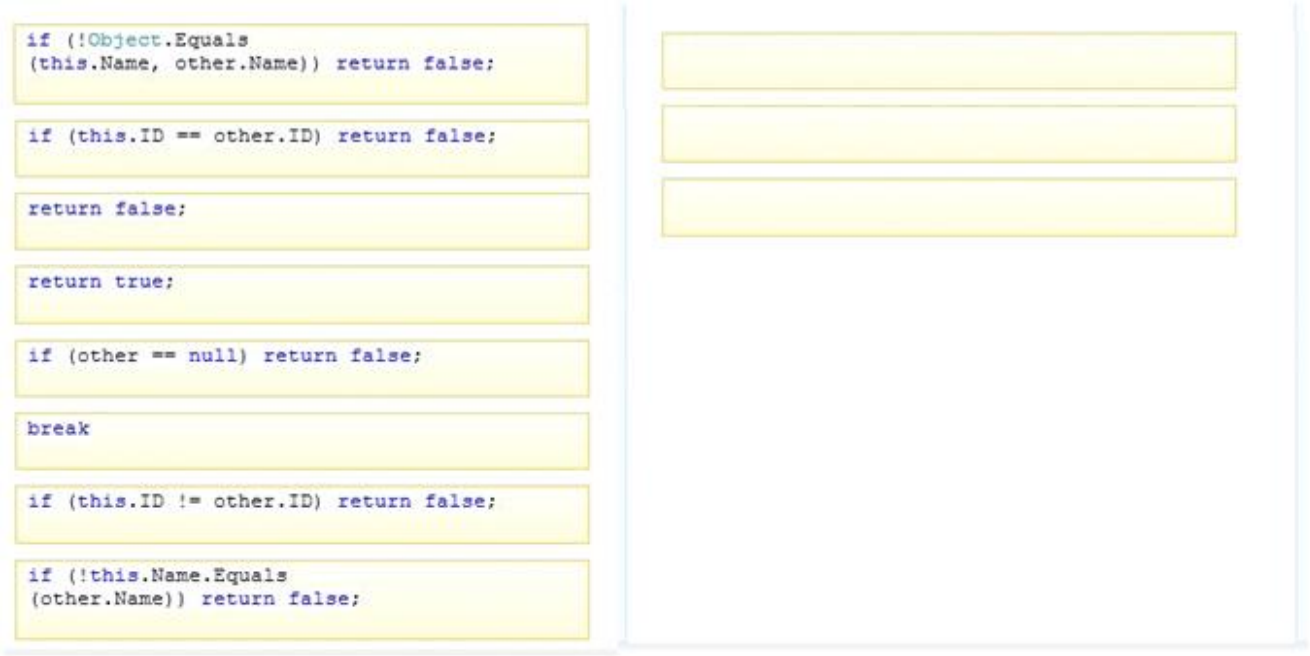
D.
```
Trace.WriteLine("Trace data...");
```

A. Option A
B. Option B
C. Option C
D. Option D
Answer: D
Explanation:
Incorrect:
Not B: There is only a "TraceListener" defined in the config file. In fact, there is no "eventlogDebugListener" class.

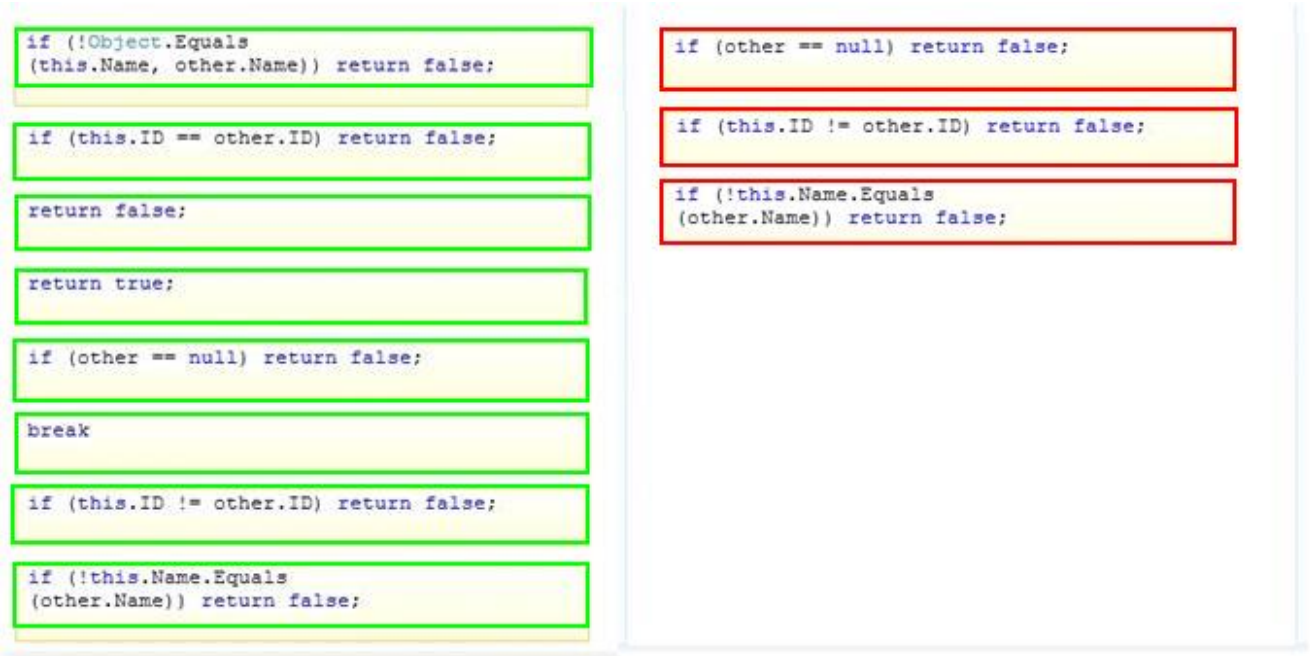
QUESTION NO: 21
あなたは次のクラスを持っています：

```
public class Class1 : IEquatable<Class1>
{
  public Int32 ID { get; set; }
  public String Name { get; set; }
  public bool Equals(Class1 other)
  {
  }
}
```

IEquatableを実装する必要があります。
IDと名前の両方が同じ値に設定されている場合、Equalsメソッドはtrueを返す必要があります。 それ以外の場合、メソッドはfalseを返す必要があります。
equalsは例外をスローしてはなりません。
あなたは何をするべきか？
（必要なコードスニペットを選択して注文することでソリューションを開発します。すべてのコードスニペットが必要なわけではありません）。



Answer:



Explanation:
In Box 3 we must use Name.Equals, not Object.Equals, to properly compare two strings.
Incorrect:
Not Box 3: Object.Equals (obj, obj) compares the REFERENCE (true if they point to same object). Two strings, even having the same value will never have the same reference. So it is not applicable here.

QUESTION NO: 22
あなたは、次のコードセグメントを含むアプリケーションを開発しています。（行番号は参考のために含まれるだけです。）

```
01 using System;
02 class MainClass
03 {
04   public static void Main(string[] args)
05   {
06     bool bValidInteger = false;
07     int value = 0;
08     do
09     {
10       Console.WriteLine("Enter an integer:");
11       bValidInteger = GetValidInteger(ref value);
12     } while (!bValidInteger);
13     Console.WriteLine("You entered a valid integer, " + value);
14   }
15   public static bool GetValidInteger(ref int val)
16   {
17     string sLine = Console.ReadLine();
18     int number;
19
20     {
21       return false;
22     }
23     else
24     {
25       val = number;
26       return true;
27     }
28   }
29 }
```

あなたは19行でどのコードセグメントを追加する必要がありますか？
Which code segment should you add at line 19?

A. If (!int.TryParse(sLine, out number))

B. If ((number = Int32.Parse(sLine)) == Single.NaN)

C. If ((number = int.Parse(sLine)) > Int32.MaxValue)

D. If (Int32.TryParse(sLine, out number))

Answer: A

Explanation:

Incorrect:

Not B, not C: These will throw exception when user enters non-integer value.

Not D: This is exactly the opposite what we want to achieve.

Int32.TryParse - Converts the string representation of a number to its 32-bit signed integer equivalent. A return value indicates whether the conversion succeeded.

http://msdn.microsoft.com/en-us/library/f02979c7.aspx

QUESTION NO: 23

あなたは、C＃を使用してアプリケーションを開発しています。

アプリケーションは、次のコードセグメントを含みます。（行番号は参考のために含まれる

だけです。)

```
01 public interface IDataContainer
02 {
03    string Data { get; set; }
04 }
05 void DoWork(object obj)
06 {
07
08    if (dataContainer != null)
09    {
10       Console.WriteLine(dataContainer.Data);
11    }
12 }
```

もしデータプロパティにアクセスする時にobjオブジェクトがタイプIDataContainerをもっていないならば、DoWork()方法はInvalidCastException例外を投げなければなりません。
あなたは、要件を満たしている必要があります。
あなたはどのコードセグメントをライン07に挿入するべきですか？

A. var dataContainer = (IDataContainer) obj;

B. var dataContainer = obj as IDataContainer;

C. var dataContainer = obj is IDataContainer;

D. dynamic dataContainer = obj;

Answer: A

Explanation:
direct cast. If object is not of the given type, an InvalidCastException is thrown.
Incorrect:
Not B: If obj is not of the given type, result is null.
Not C: If obj is not of a given type, result is false.
Not D: This simply check the variable during runtime. It will not throw an exception.

QUESTION NO: 24
アプリケーションは、反応時間を測るコードを含みます。
コードは、ユーザ・インタフェースと別の糸の上で、タイマーを動かします。アプリケーションは、以下のコードを含みます。 (行番号は参考のために含まれるだけです。)

```
01 static int RunTimer(CancellationToken cancellationToken)
02 {
03   var time = 0;
04   while (!cancellationToken.IsCancellationRequested)
05     time++;
06   return time;
07 }
08 static void Main(string[] args)
09 {
10   var tokenSource = new CancellationTokenSource();
11   var task = Task.Factory.StartNew<int>(() => RunTimer(tokenSource.Token));
12   Console.WriteLine("Press [Enter] to stop the timer.");
13   Console.ReadLine();
14
15   Console.WriteLine("Timer stopped at {0}", task.GetAwaiter().GetResult());
16   Console.ReadLine();
17 }
```

あなたは、ユーザーがエンター・キーを押すとき、アプリケーションがタイマーをキャンセルすることを確実とする必要があります。
あなたは、第14行でどのコード部分を挿入しなければなりませんか？

A. tokenSource.Token.Register(() => tokenSource.Cancel());

B. tokenSource.Cancel();

C. tokenSource.IsCancellationRequested = true;

D. tokenSource.Dispose();

Answer: B

Explanation:
The CancellationTokenSource.Cancel method communicates a request for cancellation, and specifies whether remaining callbacks and cancelable operations should be processed.
Incorrect:
Not C: The IsCancellationRequested property is ReadOnly.
Reference:
https://msdn.microsoft.com/en-us/library/dd321703(v=vs.110).aspx

QUESTION NO: 25
あなたは次のコードを持っています。（行番号は参照用にのみ記載されています）。

```
01  double x, y;
02  x = 0.0;
03  y = 0.0;
04  Console.WriteLine(x/y);
```

04行目の出力は何ですか？

A. Error

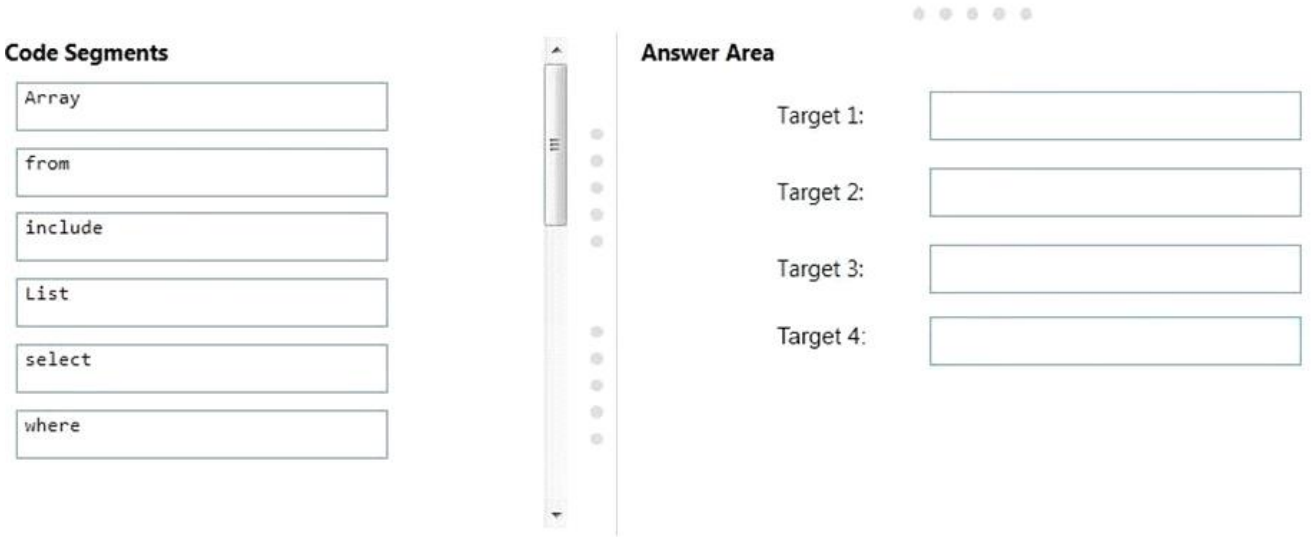
B. 0

C. null
D. NaN
Answer: B

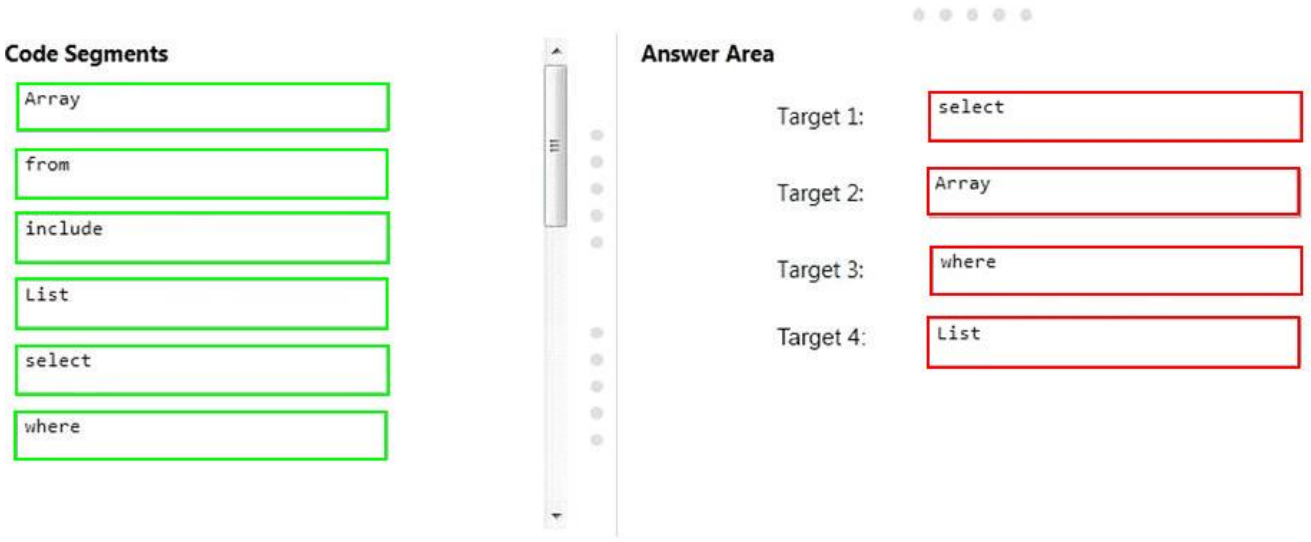
QUESTION NO: 26
次のコードがあります。

```
string[] vehicles = { "Airplane", "Boat", "Car" };
  Target 1<string> aVehicles =
  (Target 2 vehicle in vehicles
  Target 3 vehicle.StartsWith("A")
  Target 4 vehicle).ToList<string>();
    foreach (var vehicle in aVehicles)
    {
      Console.WriteLine(vehicle);
    }
```

「A」で始まるすべての車両を表示する必要があります。
どのようにコードを完成させる必要がありますか？回答するには、適切なコード要素を正しいターゲットにドラッグします。各コード要素は、1回、複数回、またはまったく使用しない場合があります。ペイン間で分割バーをドラッグするか、コンテンツを表示するにはスクロールする必要がある場合があります。
注：それぞれの正しい選択は1ポイントの価値があります。



Answer:



QUESTION NO: 27
アプリケーションには、Personという名前のクラスが含まれています。
Personクラスには、GetDataという名前のメソッドが含まれています。
GetData ( ) メソッドがPersonクラスのみで使用でき、Personクラスから派生したクラスでは使用できないようにする必要があります。
あなたはGetData()方法のためにどのアクセス変更者を使うべきであるか？
A. Public
B. Protected internal
C. Internal
D. Private
E. Protected
Answer: B
Explanation:
The protected keyword is a member access modifier. A protected member is accessible within its class and by derived class instances.

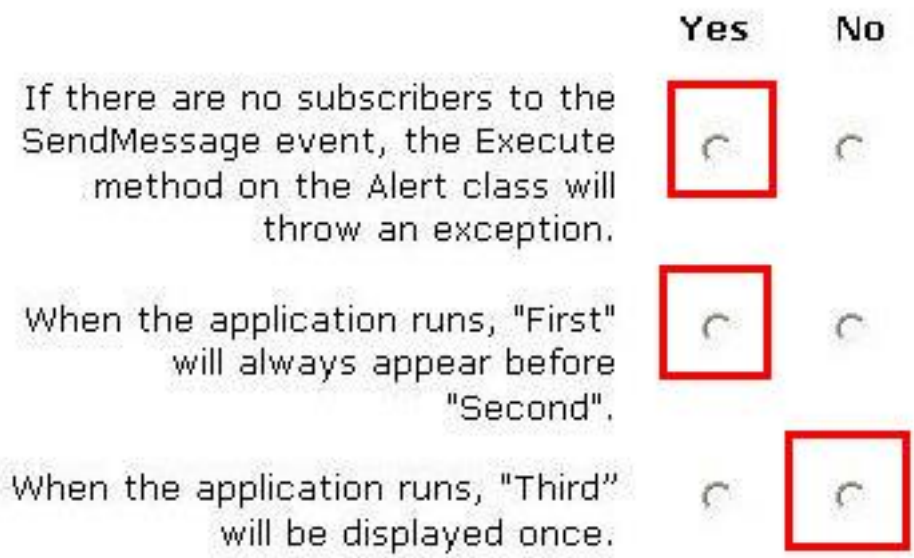
QUESTION NO: 28
あなたは次のコードを持っています：

```
public class Alert
{
  public event EventHandler<EventArgs> SendMessage;

  public void Execute()
  {
    SendMessage(this, new EventArgs());
  }
}

public class Subscriber
{
  Alert alert = new Alert();

  public void Subscribe()
  {
    alert.SendMessage += (sender, e) => { Console.WriteLine("First"); };
    alert.SendMessage += (sender, e) => { Console.WriteLine("Second"); };
    alert.SendMessage += (sender, e) => { Console.WriteLine("Third"); };
    alert.SendMessage += (sender, e) => { Console.WriteLine("Third"); };
  }

  public void Execute()
  {
    alert.Execute();
  }

  public static void Main()
  {
    Subscriber subscriber = new Subscriber();
    subscriber.Subscribe();
    subscriber.Execute();
  }
}
```

次の各文について、その文が真であればYesを選択します。 それ以外の場合は、「いいえ」

	Yes	No
If there are no subscribers to the SendMessage event, the Execute method on the Alert class will throw an exception.	○	○
When the application runs, "First" will always appear before "Second".	○	○
When the application runs, "Third" will be displayed once.	○	○

Answer:



Explanation:
Explanation for second answer: Events are multicast delegates and that one has a linked list to store the delegates in. The order of execution is always the same as they are inserted.

QUESTION NO: 29
文字列を操作する次のC＃コードがあります。

```
string str = "This is a random sentence.";

string result = str.Substring(0,str.LastIndexOf("is")) +
str.Substring(str.IndexOf("random"));
```

コードが実行したあと、結果の価値は何ですか？
A. これは文章です。
B. Thrandom randomランダムな文章。
C. これが文章です。
D. このランダムな文章。
Answer: D
Reference:
https://docs.microsoft.com/en-us/dotnet/api/system.string.substring?view=netframework-4.7.2

QUESTION NO: 30
あなたは、ファイルのためにハッシュ価値をつくるGenerateHashという名前をつけられる方法を開発しています。
方法は、以下のコードを含みます。（行番号は参考のために含まれるだけです。）

```
01 public byte[] GenerateHash(string filename, string hashAlgorithm)
02 {
03   var signatureAlgo = HashAlgorithm.Create(hashAlgorithm);
04   var fileBuffer = System.IO.File.ReadAllBytes(filename);
05
06 }
```

あなたはfileBuffer変数に含まれるバイトの暗号化ハッシュを返す必要があります。あなたはライン05でどのコードセグメントを挿入するべきですか？

A.
```
var outputBuffer = new byte[fileBuffer.Length];
signatureAlgo.TransformBlock(fileBuffer, 0, fileBuffer.Length, outputBuffer, 0);
signatureAlgo.TransformFinalBlock(fileBuffer, fileBuffer.Length - 1, fileBuffer.Length);
return outputBuffer;
```

B.
```
signatureAlgo.ComputeHash(fileBuffer);
return signatureAlgo.GetHashCode();
```

C.
```
var outputBuffer = new byte[fileBuffer.Length];
signatureAlgo.TransformBlock(fileBuffer, 0, fileBuffer.Length, outputBuffer, 0);
return outputBuffer;
```

D.
```
return signatureAlgo.ComputeHash(fileBuffer);
```

A. Option A
B. Option B
C. Option C
D. Option D
Answer: D
Explanation:
The ComputeHash(Byte[]) method computes the hash value for the specified byte array.