DUMPSBOSS.

Programming in C#

Microsoft 70-483

Version Demo

Total Demo Questions: 15

Total Premium Questions: 280

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

Topic Break Down

Topic	No. of Questions
Topic 1, Volume A	92
Topic 2, Volume B	188
Total	280

QUESTION NO: 1 - (DRAG DROP)

DRAG DROP

You need to validate whether string strJson is a valid JSON string.

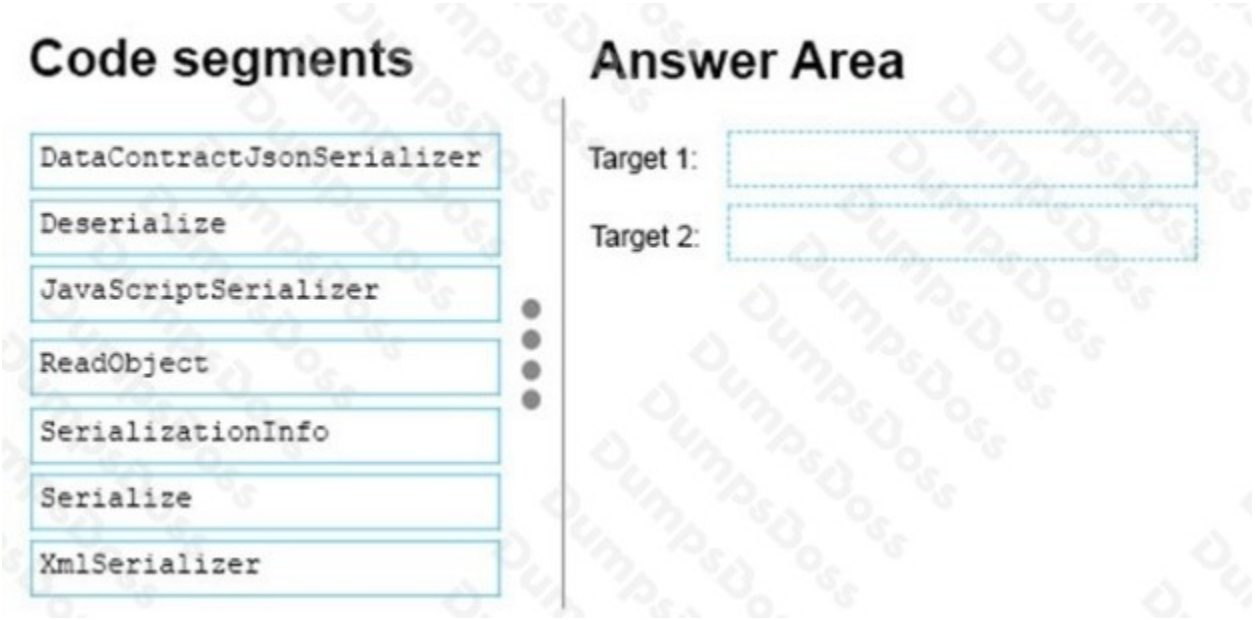
You write the following code:

```
var serializer = new Target 1();
var result = serializer.Target 2<Dictionary<string, object>>(strJson);
```

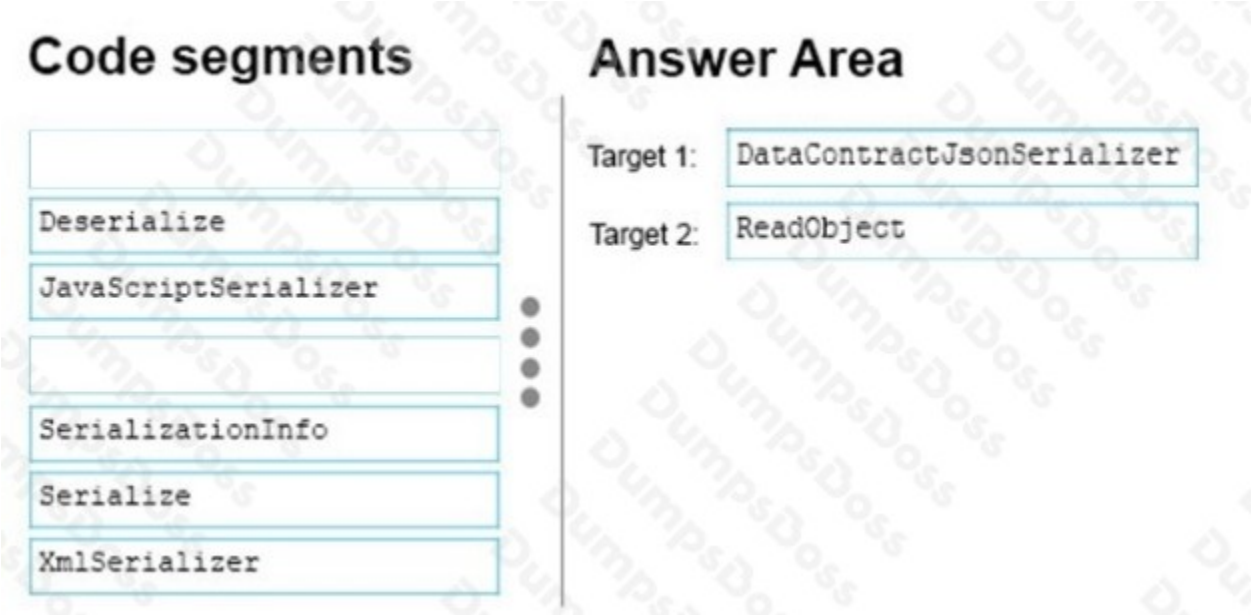
How should you complete the code? To answer, drag the appropriate code elements to the correct targets in the answer area. Each code element may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

NOTE: Each correct selection is worth one point.

Select and Place:



ANSWER:



Explanation:

serializer = new DataContractJsonSerializer(); var result = serializer.ReadObject<>>(StrJson);

QUESTION NO: 2

You are developing an application that includes the following code segment:

```
interface IHome
{
 void Start();
}
interface IOffice
{
 void Start();
}
```

You need to implement both Start() methods in a derived class named UseStart that uses the Start() method of each interface. Which two code segments should you use? (Each correct answer presents part of the solution. Choose two.)

A.
```
var starter = new UseStart();
((IHome, IOffice)starter).Start();
```

B.
```
class UseStart : IHome, IOffice
{
  public void IHome.Start()
  {
    ...
  }
  public void IOffice.Start()
  {
    ...
  }
}
```

C.
```
class UseStart : IHome, IOffice
{
  void IHome.Start()
  {
    ...
  }
  void IOffice.Start()
  {
    ...
  }
}
```

D.
```
var starter = new UseStart();
((IHome)starter).Start();
((IOffice)starter).Start();
```

E.
```
var starter = new UseStart();
starter.Start(IHome);
starter.Start(IOffice);
```

F.
```
var starter = new UseStart();
starter.Start();
```

A. Option A

B. Option B

C. Option C

D. Option D

E. Option E

F. Option F

ANSWER: C D

Explanation:

C:

Implementing Multiple Interfaces

A class can implement multiple interfaces using the following syntax:

C#

public class CDAndDVDComboPlayer : ICDPlayer, IDVDPlayer

If a class implements more than one interface where there is ambiguity in the names of members, it is resolved using the full qualifier for the property or method name. In other words, the derived class can resolve the conflict by using the fully qualified name for the method to indicate to which interface it belongs

In C#, both inheritance and interface implementation are defined by the : operator, equivalent to extends and implements in Java. The base class should always be leftmost in the class declaration.

QUESTION NO: 3

You are creating a console application by using C#.

You need to access the application assembly.

Which code segment should you use?

A. Assembly.GetAssembly(this);

B. this.GetType();

C. Assembly.Load();

D. Assembly.GetExecutingAssembly();

ANSWER: D

Explanation:

▪ Assembly.GetExecutingAssembly - Gets the assembly that contains the code that is currently executing. ▪ Assembly.GetAssembly - Gets the currently loaded assembly in which the specified class is defined.

References:

http://msdn.microsoft.com/en-us/library/system.reflection.assembly.getassembly.aspx http://msdn.microsoft.com/en-us/library/system.reflection.assembly.getexecutingassembly(v=vs.110).aspx

QUESTION NO: 4 - (DRAG DROP)

DRAG DROP

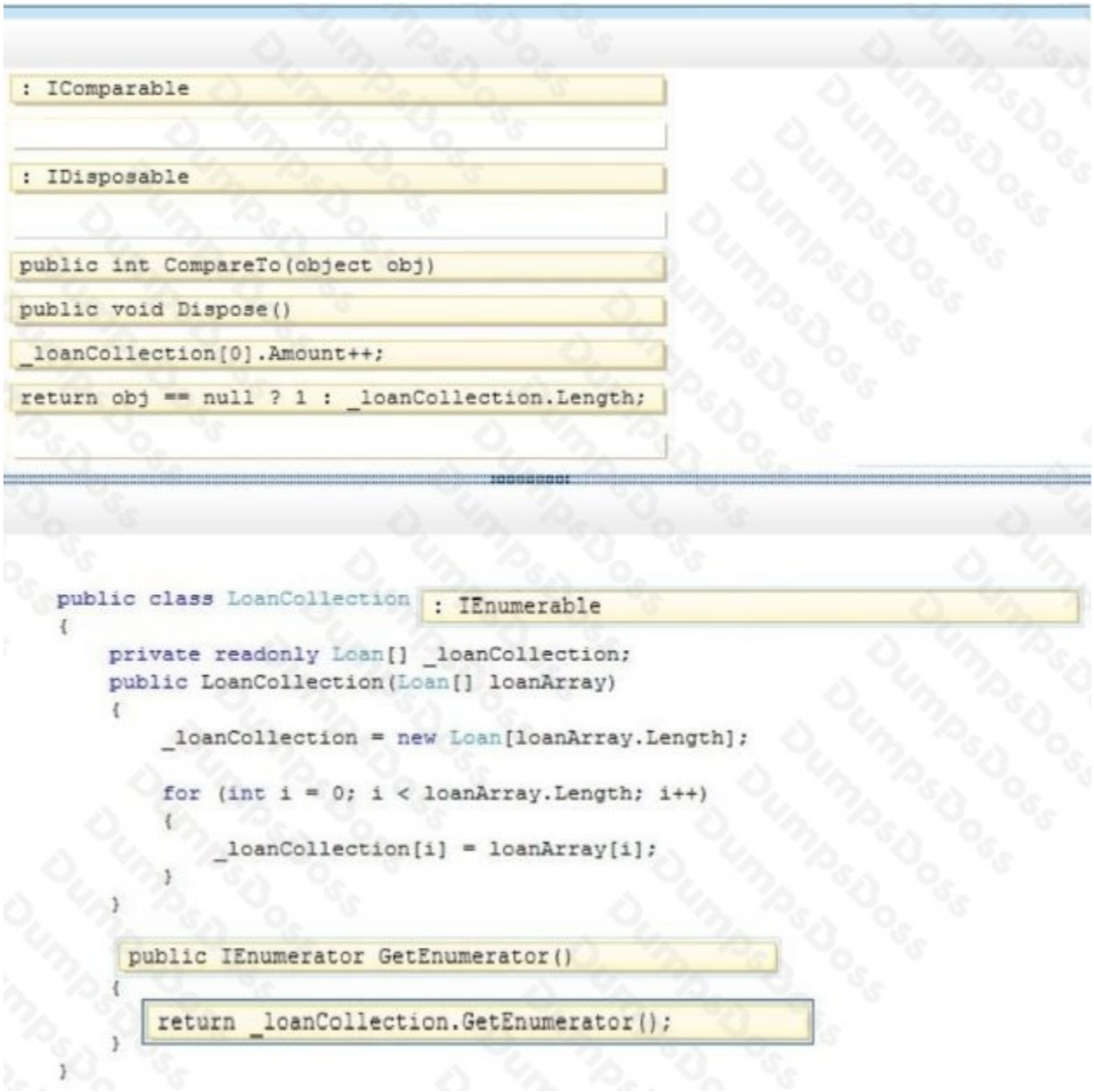
You are developing a custom collection named LoanCollection for a class named Loan class.

You need to ensure that you can process each Loan object in the LoanCollection collection by using a foreach loop.

How should you complete the relevant code? (To answer, drag the appropriate code segments to the correct locations in the answer area. Each code segment may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.)

Select and Place:

ANSWER:



QUESTION NO: 5

You develop an application by using C#. The application counts the number of times a specific word appears within a set of text files. The application includes the following code. (Line numbers are included for reference only.)

```
01 class Counter
02 {
03   System.Collections.Concurrent.ConcurrentDictionary<string, int> _wordCounts =
04     new System.Collections.Concurrent.ConcurrentDictionary<string, int>();
05   public Action<DirectoryInfo> ProcessDirectory()
06   {
07     return (dirInfo =>
08     {
09       var files = dirInfo.GetFiles("*.cs").AsParallel<FileInfo>();
10       files.ForAll<FileInfo>(
11        fileInfo =>
12        {
13          var fileContent = File.ReadAllText(fileInfo.FullName);
14          var sb = new StringBuilder();
15          foreach (var val in fileContent)
16          {
17            sb.Append(char.IsLetter(val) ? val.ToString().ToLowerInvariant() : " ");
18          }
19          string[] wordsInFile = sb.ToString().Split(new []{' '},
20           StringSplitOptions.RemoveEmptyEntries);
21          foreach (var word in wordsInFile)
22          {
23
24          }
25        });
26       var directories = dirInfo.GetDirectories().AsParallel<DirectoryInfo>();
27       directories.ForAll<DirectoryInfo>(ProcessDirectory());
28     });
29   }
30 }
```

You have the following requirements:

▪ Populate the _wordCounts object with a list of words and the number of occurrences of each word. ▪ Ensure that updates to the ConcurrentDictionary object can happen in parallel.

You need to complete the relevant code.

Which code segment should you insert at line 23?

A.
```
_wordCounts.AddOrUpdate(word, 1, (s, n) => n + 1);
```

B.
```
int value;
if (_wordCounts.TryGetValue(word, out value))
{
  _wordCounts[word] = value++;
}
else
{
  _wordCounts[word] = 1;
}
```

C.
```
var value = _wordCounts.GetOrAdd(word, 0);
_wordCounts[word] = value++;
```

D.
```
var value = _wordCounts.GetOrAdd(word, 0);
_wordCounts.TryUpdate(word, value + 1, value);
```

A. Option A

B. Option B

C. Option C

D. Option D

ANSWER: A

QUESTION NO: 6

You are developing an application that includes a method named SendMessage.

You need to ensure that the SendMessage() method is called with the required parameters.

Which two code segments can you use to achieve this goal? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

A.
```
static void Main(string[] args)
{
  dynamic message = new { From = "Jon Morris", To = "Mary North", Content = "Hello World" };
  SendMessage(message);
}
private static void SendMessage(Object msg)
{
  Console.WriteLine(msg.From);
  Console.WriteLine(msg.To);
  Console.WriteLine(msg.Content);
}
```

B.
```
static void Main(string[] args)
{
  var message = new Object();
  message.From = "Jon Morris";
  message.To = "Mary North";
  message.Content = "Hello World";
  SendMessage(message);
}
private static void SendMessage(dynamic msg)
{
  Console.WriteLine(msg.From);
  Console.WriteLine(msg.To);
  Console.WriteLine(msg.Content);
}
```

C.
```
static void Main(string[] args)
{
  var message = new { From = "Jon Morris", To = "Mary North", Content = "Hello World" };
  SendMessage(message);
}
private static void SendMessage(dynamic msg)
{
  Console.WriteLine(msg.From);
  Console.WriteLine(msg.To);
  Console.WriteLine(msg.Content);
}
```

D.
```
static void Main(string[] args)
{
  dynamic message = new ExpandoObject();
  message.From = "Jon Morris";
  message.To = "Mary North";
  message.Content = "Hello World";
  SendMessage(message);
}
private static void SendMessage(dynamic msg)
{
  Console.WriteLine(msg.From);
  Console.WriteLine(msg.To);
  Console.WriteLine(msg.Content);
}
```

A. Option A

B. Option B

C. Option C

D. Option D

ANSWER: C D

Explanation:

D: ExpandoObject

Represents an object whose members can be dynamically added and removed at run time.

▪ The ExpandoObject class enables you to add and delete members of its instances at run time and also to set and get values of these members. This class supports dynamic binding, which enables you to use standard syntax like sampleObject.sampleMember instead of more complex syntax like sampleObject.GetAttribute("sampleMember").

▪ You can pass instances of the ExpandoObject class as parameters. Note that these instances are treated as dynamic objects in C# and late-bound objects in Visual Basic. This means that you do not have IntelliSense for object members and you do not receive compiler errors when you call non-existent members. If you call a member that does not exist, an exception occurs.

Note:

Visual C# 2010 introduces a new type, dynamic. The type is a static type, but an object of type dynamic bypasses static type checking. In most cases, it functions like it has type object. At compile time, an element that is typed as dynamic is assumed to support any operation. Therefore, you do not have to be concerned about whether the object gets its value from a COM API, from a dynamic language such as IronPython, from the HTML Document Object Model (DOM), from reflection, or from somewhere else in the program. However, if the code is not valid, errors are caught at run time.

QUESTION NO: 7

You are developing an application that includes the following code segment. (Line numbers are included for reference only.)

```
01 class Customer
02 {
03   public string CompanyName { get; set; }
04   public string Id { get; set; }
05 }
06 const string sqlSelectCustomers = "SELECT CustomerID, CompanyName FROM Customers";
07 private static IEnumerable<Customer> GetCustomers(string sqlConnectionString)
08 {
09   List<Customer> customers = new List<Customer>();
10   SqlConnection sqlConnection = new SqlConnection(sqlConnectionString);
11   using (sqlConnection)
12   {
13     SqlCommand sqlCommand = new SqlCommand(sqlSelectCustomers, sqlConnection);
14
15     using (SqlDataReader sqlDataReader = sqlCommand.ExecuteReader())
16     {
17
18       {
19         Customer customer = new Customer();
20         customer.Id = (string)sqlDataReader["CustomerID"];
21         customer.CompanyName = (string)sqlDataReader["CompanyName"];
22         customers.Add(customer);
23       }
24     }
25   }
26   return customers;
27 }
```

The GetCustomers() method must meet the following requirements:

▪ Connect to a Microsoft SQL Server database.

▪ Populate Customer objects with data from the database.

▪ Return an IEnumerable collection that contains the populated Customer objects.

You need to meet the requirements.

Which two actions should you perform? (Each correct answer presents part of the solution. Choose two.)

A. Insert the following code segment at line 17: while (sqlDataReader.GetValues())

B. Insert the following code segment at line 14: sqlConnection.Open();

C. Insert the following code segment at line 14: sqlConnection.BeginTransaction();

D. Insert the following code segment at line 17: while (sqlDataReader.Read())

E. Insert the following code segment at line 17: while (sqlDataReader.NextResult())

ANSWER: B D

Explanation:

SqlConnection.Open - Opens a database connection with the property settings specified by the ConnectionString.
SqlDataReader.Read - Advances the SqlDataReader to the next record.

References:

http://msdn.microsoft.com/en-us/library/system.data.sqlclient.sqlconnection.open.aspx http://msdn.microsoft.com/en-us/library/system.data.sqlclient.sqldatareader.read.aspx

QUESTION NO: 8

You are developing an application by using C#. You provide a public key to the development team during development.

You need to specify that the assembly is not fully signed when it is built.

Which two assembly attributes should you include in the source code? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

A. AssemblyKeyNameAttribute

B. ObfuscateAssemblyAttribute

C. AssemblyDelaySignAttribute

D. AssemblyKeyFileAttribute

E. AssemblyFlagsAttribute

F. AssemblyConfigurationAttribute

ANSWER: C D

Explanation:

References: http://msdn.microsoft.com/en-us/library/t07a3dye(v=vs.110).aspx

QUESTION NO: 9

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.

After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You have the following C# code. (Line numbers are included for reference only.)

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

You need the foreach loop to display a running total of the array elements, as shown in the following list.

1

3

6

10

15

Solution: You insert the following code segment at line 02:

```
int sum = 0;
for (int i=0; i < intArray.Lenght;)
{
   sum += intArray[i];
   intArray[i++] = sum;
   Console.WriteLine(sum);
}
```

Does this meet the goal?

A. Yes

B. No

ANSWER: B

Explanation:

Console.WriteLine is repeated twice.

QUESTION NO: 10

You have the following class definition.

```
public class ProcessManagement
{
    public int DegreeOfParallelism;
    private int NumberOfTasks = 0;
    public void SpawnTasks()
    {
        if (DegreeOfParallelism>20) { DegreeOfParallelism = 20};
        while (NumberOfTasks != DegreeOfParallelism)
        {
            CreateNewTask();
            NumberOfTasks++;
        }
    }
}
```

You discover that when you execute the following code, the SpawnTasks method enters an infinite loop.

```
ProcessManagement pm = new ProcessManagement();
pm.DegreeOfParallelism = -1;
pm.SpawnTasks();
```

You need to prevent the SpawnTasks method from entering an infinite loop.

Which two changes should you make to the code? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

A. Add a property to the ProcessManagement class. Modify the property to allow only positive values to be stored in the DegreeOfParallelism member variable.

B. Add a property to the ProcessManagement class. Modify the property to allow only positive values to be stored in the NumberOfTasks member variable.

C. Change the accessor of the ProcessManagement class to internal.

D. Change the accessor of the DegreeOfParallelism member variable to private.

E. Change the accessor of the SpawnTasks method to private.

ANSWER: A B

QUESTION NO: 11

You are developing a C# application. The application references and calls a RESTful web service named EmployeeService. The EmployeeService web service includes a method named GetEmployee, which accepts an employee ID as a parameter. The web service returns the following JSON data from the method.

{"Id":1,"Name":"David Jones"}

The following code segment invokes the service and stores the result:

```
WebClient client = new WebClient();
byte[] employeeData = client.DownloadData("http://localhost:2588/EmployeeService.svc/GetEmployee/1");
```

You need to convert the returned JSON data to an Employee object for use in the application.

Which code segment should you use?

A.
```
using (Stream stream = new MemoryStream(employeeData))
{
  XmlSerializer xmlSerializer = new XmlSerializer(typeof(Employee));
  Employee retrievedEmployee = xmlSerializer.Deserialize(stream) as Employee;
  ...
}
```

B.
```
using (Stream stream = new MemoryStream(employeeData))
{
  DataContractSerializer dataContractSerializer = new DataContractSerializer(typeof(Employee));
  Employee retrievedEmployee = dataContractSerializer.ReadObject(stream) as Employee;
  ...
}
```

C.
```
using (Stream stream = new MemoryStream(employeeData))
{
  DataContractJsonSerializer dataContractJsonSerializer = new DataContractJsonSerializer(typeof(Employee));
  Employee retrievedEmployee = dataContractJsonSerializer.ReadObject(stream) as Employee;
  ...
}
```

D.
```
using (Stream stream = new MemoryStream(employeeData))
{
  NetDataContractSerializer netDataContractSerializer = new NetDataContractSerializer();
  Employee retrievedEmployee = netDataContractSerializer.ReadObject(stream) as Employee;
  ...
}
```

A. Option A

B. Option B

C. Option C

D. Option D

ANSWER: C

QUESTION NO: 12

You are developing an application that uses multiple asynchronous tasks to optimize performance. The application will be deployed in a distributed environment.

You need to retrieve the result of an asynchronous task that retrieves data from a web service. The data will be later being parsed by a separate task.

Which code segment should you use?

A.
```
protected async void StartTask()
{
  string result = await GetData();
  ...
}
public Task<string> GetData()
{
  ...
}
```

B.
```
protected async void StartTask()
{
  string result = GetData();
  ...
}
public Task<string> GetData()
{
  ...
}
```

C.
```
protected async void StartTask()
{
  string result = await GetData();
  ...
}
public async Task<string> GetData()
{
  ...
}
```

D.
```
protected async void StartTask()
{
  string result = async GetData();
  ...
}
public await Task<string> GetData()
{
  ...
}
```

A. Option A

B. Option B

C. Option C

D. Option D

ANSWER: C

QUESTION NO: 13

You are developing an application.

The application contains the following code:

```
class Program
{
  static void ProcessOrders (string orderRefNumber)
  {
    if (orderRefNumber == null)
    {
      throw new ArgumentNullException();
    }
    ...
  }

  static void Main()
  {
    try
    {
      string orderRefNumber = null;
      ProcessOrders(orderRefNumber);
    }
    catch (ArgumentNullException e)
    {
      Console.WriteLine("{0} An exception caught.", e);
    }

    catch (Exception e)
    {
      Console.WriteLine("{0} An exception caught.", e);
    }
  }
}
```

When you compile the code, you receive the following syntax error message: "A previous catch clause already catches all exceptions of this or a super type ('System.Exception')." You need to ensure that the code can be compiled. What should you do?

A. Catch the ArgumentException exception instead of the ArgumentNullException exception.

B. Throw a new exception in the second catch block.

C. Catch the ArgumentNullException exception first.

D. Re-throw the exception caught by the second catch block.

ANSWER: A

QUESTION NO: 14 - (HOTSPOT)

HOTSPOT

You are reviewing the following code:

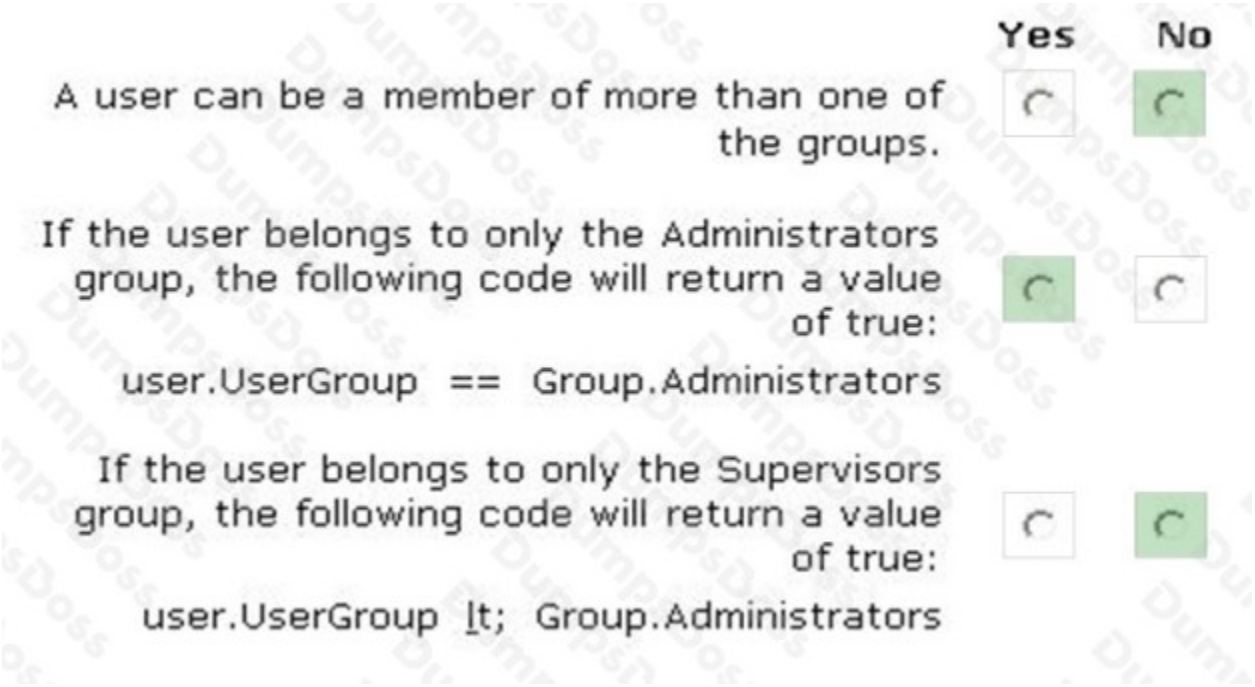
```
[System.FlagsAttribute()]
public enum Group
{
  Users = 1,
  Supervisors = 2,
  Managers = 4,
  Administrators = 8
}
public class User
{
  public Group UserGroup { get; set; }
}
```

For each of the following statements, select Yes if the statement is true. Otherwise, select No.

Hot Area:

	Yes	No
A user can be a member of more than one of the groups.	○	○
If the user belongs to only the Administrators group, the following code will return a value of true: `user.UserGroup == Group.Administrators`	○	○
If the user belongs to only the Supervisors group, the following code will return a value of true: `user.UserGroup lt; Group.Administrators`	○	○

ANSWER:



QUESTION NO: 15 - (HOTSPOT)

HOTSPOT

You have the following code:

```
private static Dictionary<string, int> CreateTestData()
{
    Dictionary<string, int> dict = new Dictionary<string, int>()
    {
        {"Accounting", 1},
        {"Marketing",  2},
        {"Operations", 3},
    }
    return dict;
}
private static bool? FindInList(string searchTerm, int value)
{
    Dictionary<string, int> data = CreateTestData();
    return data.Contains(new KeyValuePair<string,int>(searchTerm,value));
}
```

Use the drop-down lists to select the answer choice that completes each statement.

Hot Area:

If the search term is set to "Finance", and value is set to 0, the result will be **[answer choice]**.

If the search term is set to "Accounting", and value is set to 1, the result will be **[answer choice]**.

If the search term is set to "Accounting", and value is set to 2, the result will be **[answer choice]**.

ANSWER:

If the search term is set to "Finance", and value is set to 0, the result will be **[answer choice]**.

If the search term is set to "Accounting", and value is set to 1, the result will be **[answer choice]**.

If the search term is set to "Accounting", and value is set to 2, the result will be **[answer choice]**.