考試編碼:70-483

考試名稱: Programming in C#

版本: Demo

Q 1

You are developing an application that includes a class named Order. The application will store a collection of Order objects.

The collection must meet the following requirements:

Use strongly typed members.
Process Order objects in first-in-first-out order.
Store values for each Order object.
Use zero-based indices.

You need to use a collection type that meets the requirements.

Which collection type should you use?

A. Queue<T>
B. SortedList
C. LinkedList<T>
D. HashTable
E. Array<T>

Answer: A

Q 2

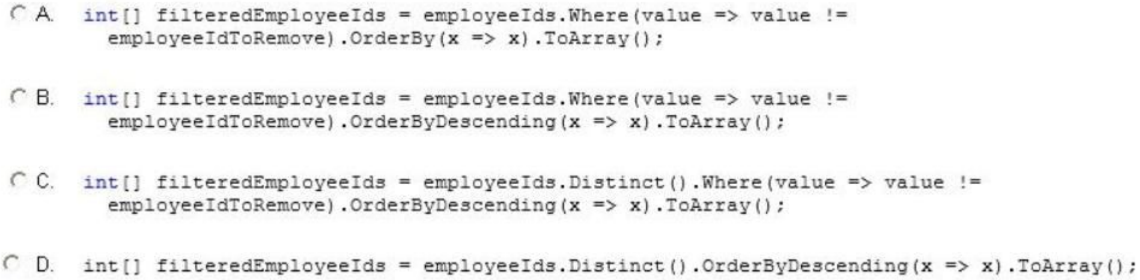
You are developing an application. The application calls a method that returns an array of integers named employeeIds. You define an integer variable named employeeIdToRemove and assign a value to it. You declare an array named filteredEmployeeIds.

You have the following requirements:

Remove duplicate integers from the employeeIds array. Sort the array in order from the highest value to the lowest value. Remove the integer value stored in the employeeIdToRemove variable from the employeeIds array.

You need to create a LINQ query to meet the requirements.

Which code segment should you use?



```
○ A.  int[] filteredEmployeeIds = employeeIds.Where(value => value !=
        employeeIdToRemove).OrderBy(x => x).ToArray();

○ B.  int[] filteredEmployeeIds = employeeIds.Where(value => value !=
        employeeIdToRemove).OrderByDescending(x => x).ToArray();

○ C.  int[] filteredEmployeeIds = employeeIds.Distinct().Where(value => value !=
        employeeIdToRemove).OrderByDescending(x => x).ToArray();

○ D.  int[] filteredEmployeeIds = employeeIds.Distinct().OrderByDescending(x => x).ToArray();
```

A. Option A

B. Option B

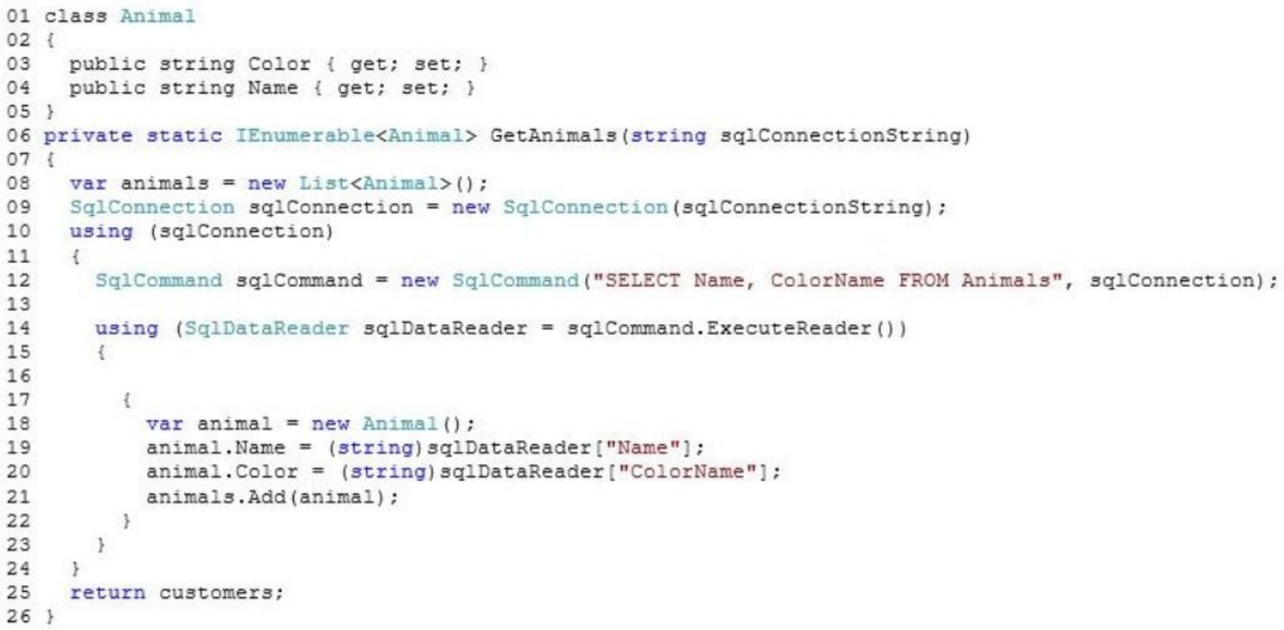
C. Option C

D. Option D

Answer: C

Q 3

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



```
01 class Animal
02 {
03   public string Color { get; set; }
04   public string Name { get; set; }
05 }
06 private static IEnumerable<Animal> GetAnimals(string sqlConnectionString)
07 {
08   var animals = new List<Animal>();
09   SqlConnection sqlConnection = new SqlConnection(sqlConnectionString);
10   using (sqlConnection)
11   {
12     SqlCommand sqlCommand = new SqlCommand("SELECT Name, ColorName FROM Animals", sqlConnection);
13
14     using (SqlDataReader sqlDataReader = sqlCommand.ExecuteReader())
15     {
16
17       {
18         var animal = new Animal();
19         animal.Name = (string)sqlDataReader["Name"];
20         animal.Color = (string)sqlDataReader["ColorName"];
21         animals.Add(animal);
22       }
23     }
24   }
25   return customers;
26 }
```

The GetAnimals() method must meet the following requirements:

Connect to a Microsoft SQL Server database.

Create Animal objects and populate them with data from the database.

Return a sequence of populated Animal 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 16:
while (sqlDataReader.NextResult())
B. Insert the following code segment at line 13:
sqlConnection.BeginTransaction();
C. Insert the following code segment at line 13:
sqlConnection.Open();
D. Insert the following code segment at line 16:
while (sqlDataReader.Read())
E. insert the following code segment at line 16:
while (sqlDataReader.GetValues())

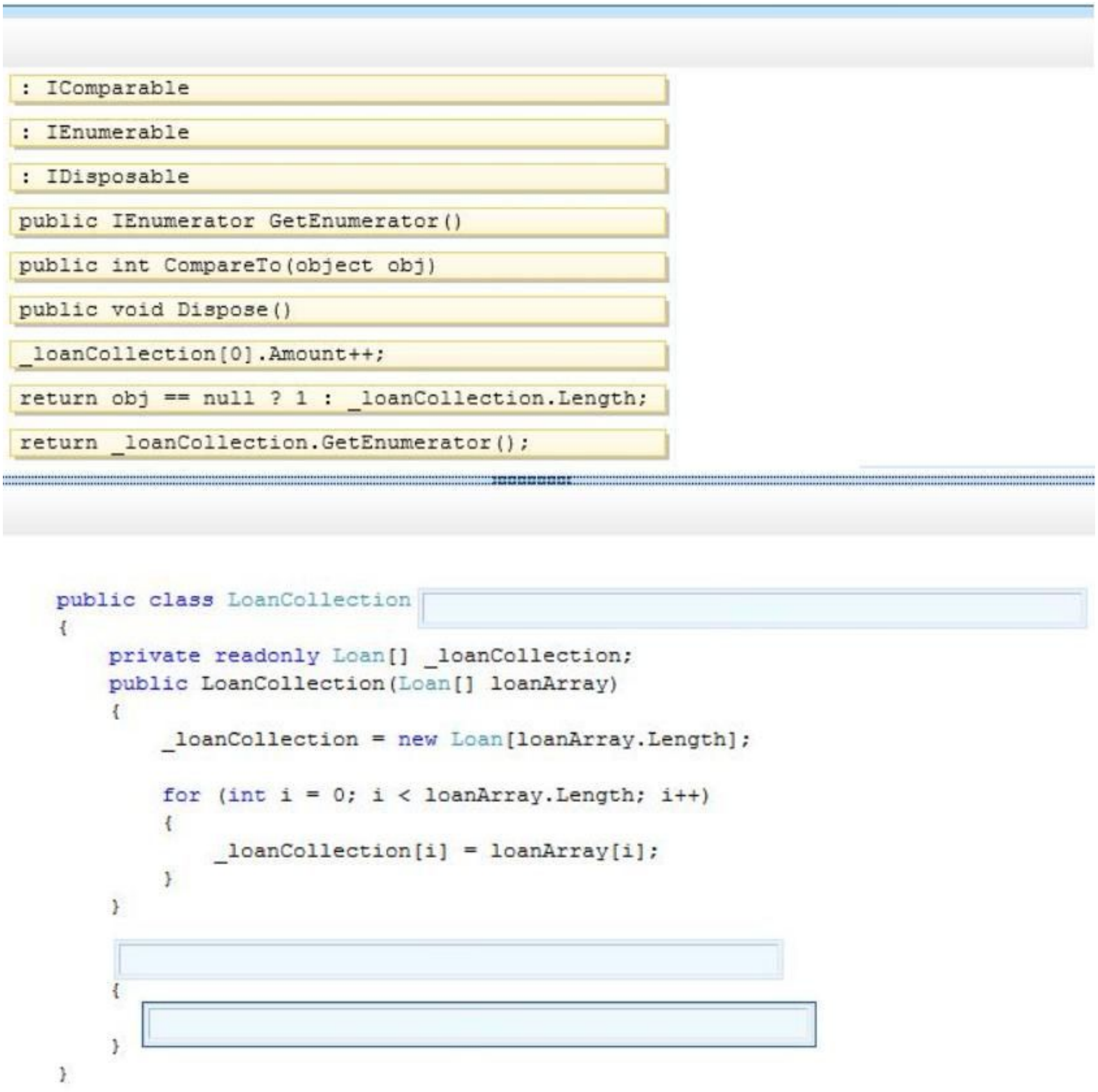
Answer: C,D

Q 4
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.)



```
: IComparable
```

```
: IEnumerable
```

```
: IDisposable
```

```
public IEnumerator GetEnumerator()
```

```
public int CompareTo(object obj)
```

```
public void Dispose()
```

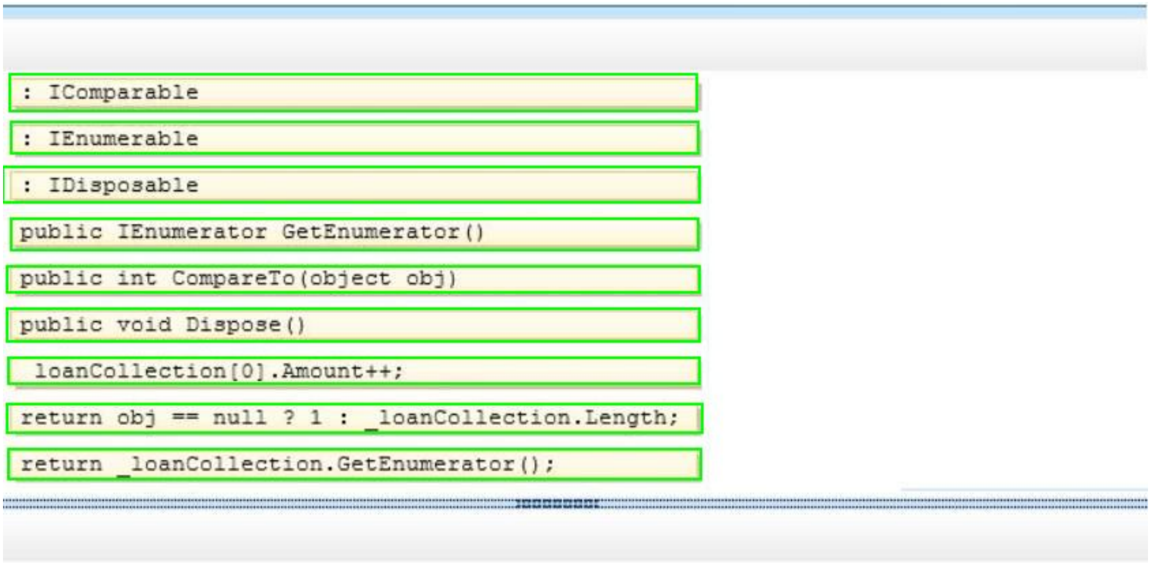
```
_loanCollection[0].Amount++;
```

```
return obj == null ? 1 : _loanCollection.Length;
```

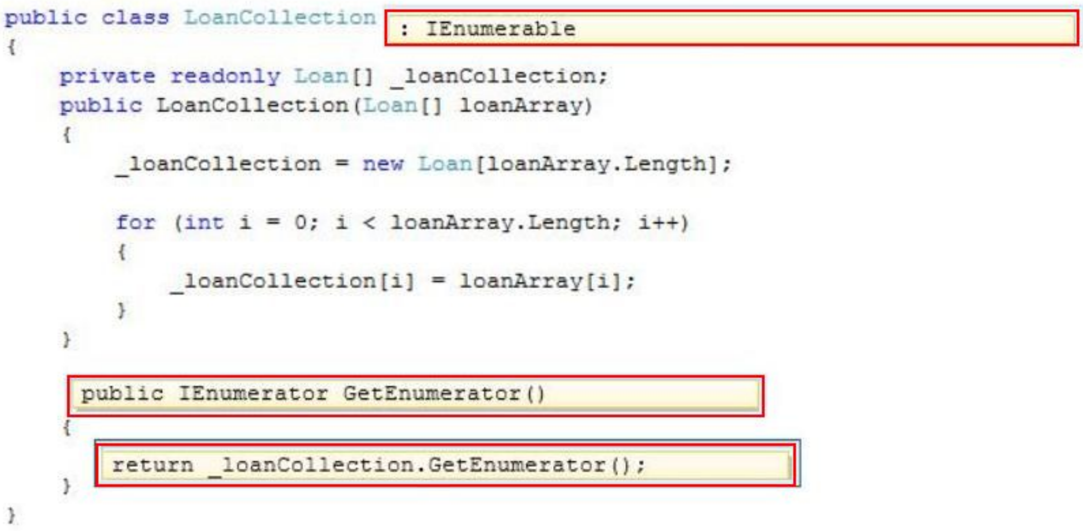
```
return _loanCollection.GetEnumerator();
```

```
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:



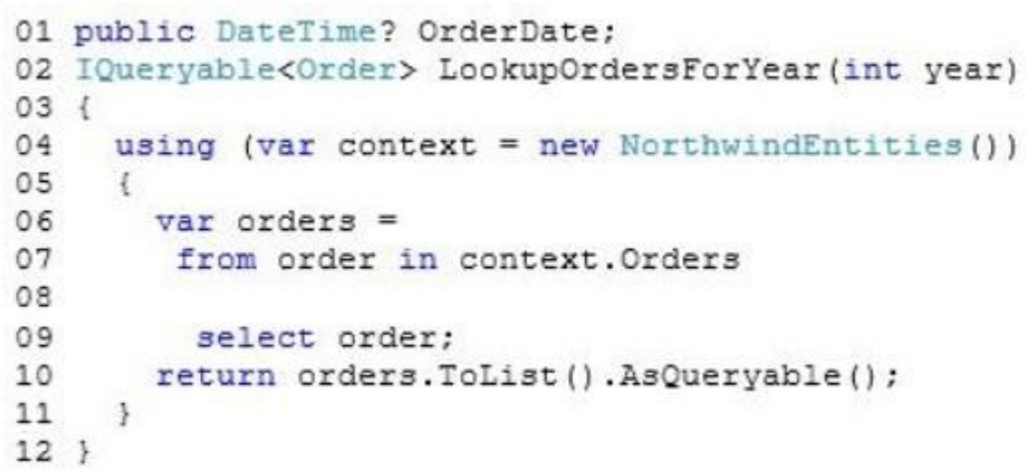
```
: IComparable
: IEnumerable
: IDisposable
public IEnumerator GetEnumerator()
public int CompareTo(object obj)
public void Dispose()
 loanCollection[0].Amount++;
return obj == null ? 1 : _loanCollection.Length;
return _loanCollection.GetEnumerator();
```



```
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();
    }
}
```

Q 5

You are developing an application that uses the Microsoft ADO.NET Entity Framework to retrieve order information from a Microsoft SQL Server database. The application includes the following code. (Line numbers are included for reference only.)



```
01 public DateTime? OrderDate;
02 IQueryable<Order> LookupOrdersForYear(int year)
03 {
04   using (var context = new NorthwindEntities())
05   {
06     var orders =
07      from order in context.Orders
08
09       select order;
10     return orders.ToList().AsQueryable();
11   }
12 }
```

The application must meet the following requirements:

Return only orders that have an OrderDate value other than null. Return only orders that were placed in the year specified in the OrderDate property or in a later year.

You need to ensure that the application meets the requirements.

Which code segment should you insert at line 08?

A. Where order.OrderDate.Value != null && order.OrderDate.Value.Year > = year
B. Where order.OrderDate.Value = = null && order.OrderDate.Value.Year = = year
C. Where order.OrderDate.HasValue && order.OrderDate.Value.Year = = year
D. Where order.OrderDate.Value.Year = = year

Answer: A

Explanation: *For the requirement to use an OrderDate value other than null use:
OrderDate.Value != null

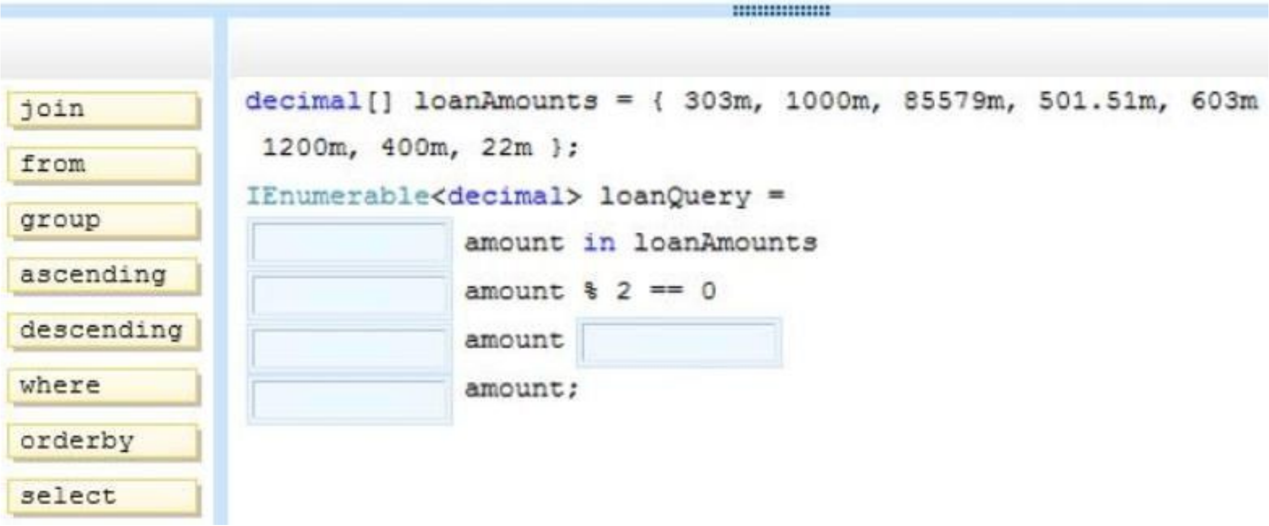
*For the requirement to use an OrderDate value for this year or a later year use:
OrderDate.Value>= year

Q 6
DRAG DROP
You are developing an application by using C#. The application includes an array of decimal values named loanAmounts. You are developing a LINQ query to return the values from the array.

The query must return decimal values that are evenly divisible by two. The values must be sorted from the lowest value to the highest value.

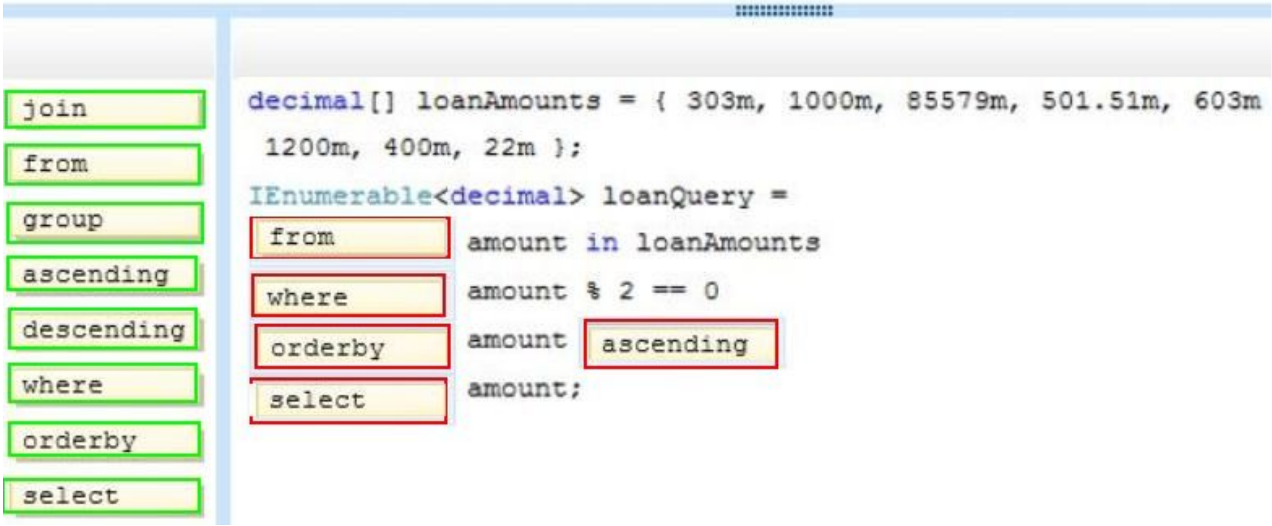
You need to ensure that the query correctly returns the decimal values.

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.)



```
decimal[] loanAmounts = { 303m, 1000m, 85579m, 501.51m, 603m
 1200m, 400m, 22m };
IEnumerable<decimal> loanQuery =
            amount in loanAmounts
            amount % 2 == 0
            amount
            amount;
```

Answer:



```
decimal[] loanAmounts = { 303m, 1000m, 85579m, 501.51m, 603m
 1200m, 400m, 22m };
IEnumerable<decimal> loanQuery =
from        amount in loanAmounts
where       amount % 2 == 0
orderby     amount ascending
select      amount;
```

Q 7

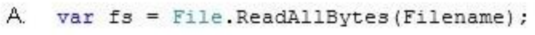
You are developing an application. The application includes a method named ReadFile that reads data from a file.

The ReadFile() method must meet the following requirements:

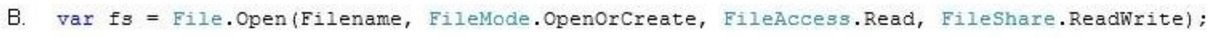
It must not make changes to the data file.
It must allow other processes to access the data file. It must not throw an exception if the application attempts to open a data file that does not exist.

You need to implement the ReadFile() method.

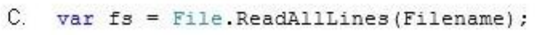
Which code segment should you use?



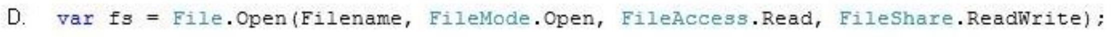
```
A.  var fs = File.ReadAllBytes(Filename);
```



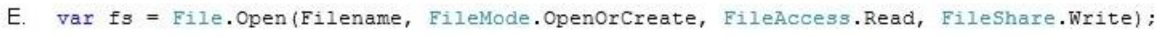
```
B.  var fs = File.Open(Filename, FileMode.OpenOrCreate, FileAccess.Read, FileShare.ReadWrite);
```



```
C.  var fs = File.ReadAllLines(Filename);
```



```
D.  var fs = File.Open(Filename, FileMode.Open, FileAccess.Read, FileShare.ReadWrite);
```

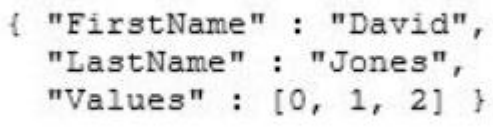


```
E.  var fs = File.Open(Filename, FileMode.OpenOrCreate, FileAccess.Read, FileShare.Write);
```

A. Option A
B. Option B
C. Option C
D. Option D
E. Option E

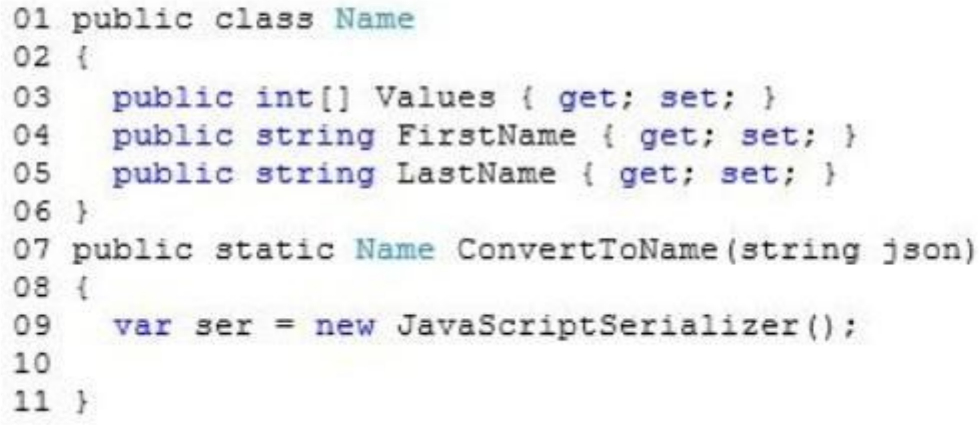
Answer: B

Q 8
An application receives JSON data in the following format:



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

The application includes the following code segment. (Line numbers are included for reference only.)



```
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 }
```

You need to ensure that the ConvertToName() method returns the JSON input string as a Name object.

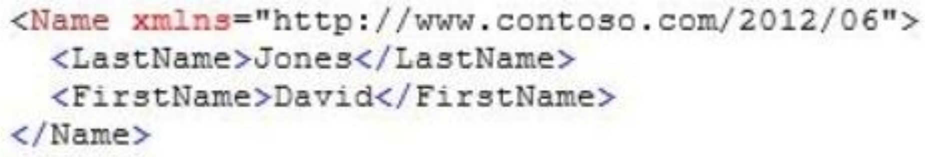
Which code segment should you insert at line 10?

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

Q 9
DRAG DROP

An application serializes and deserializes XML from streams. The XML streams are in the following format:



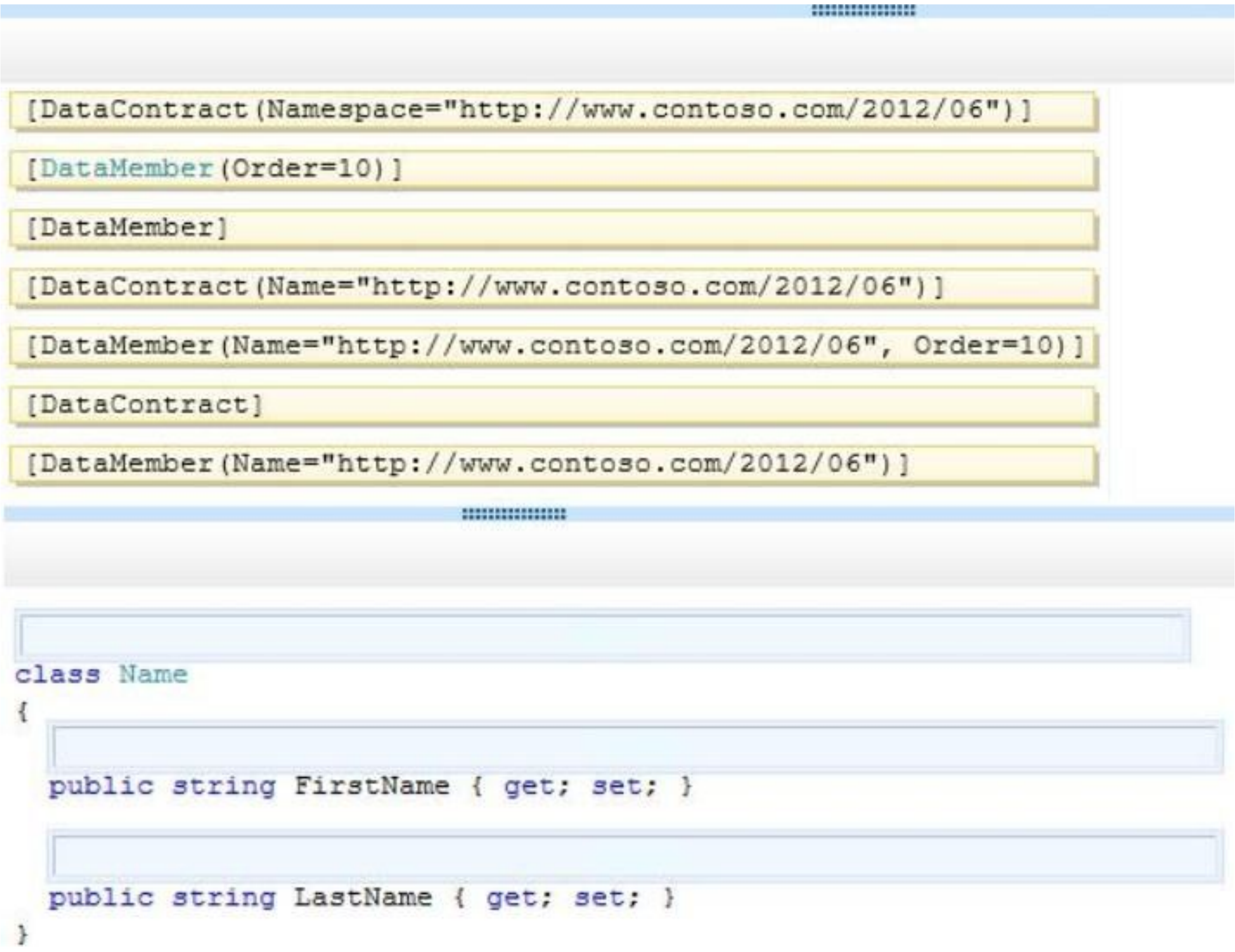
```
<Name xmlns="http://www.contoso.com/2012/06">
  <LastName>Jones</LastName>
  <FirstName>David</FirstName>
</Name>
```

The application reads the XML streams by using a DataContractSerializer object that is declared by the following code segment:

var ser = new DataContractSerializer(typeof(Name));

You need to ensure that the application preserves the element ordering as provided in the XML stream.

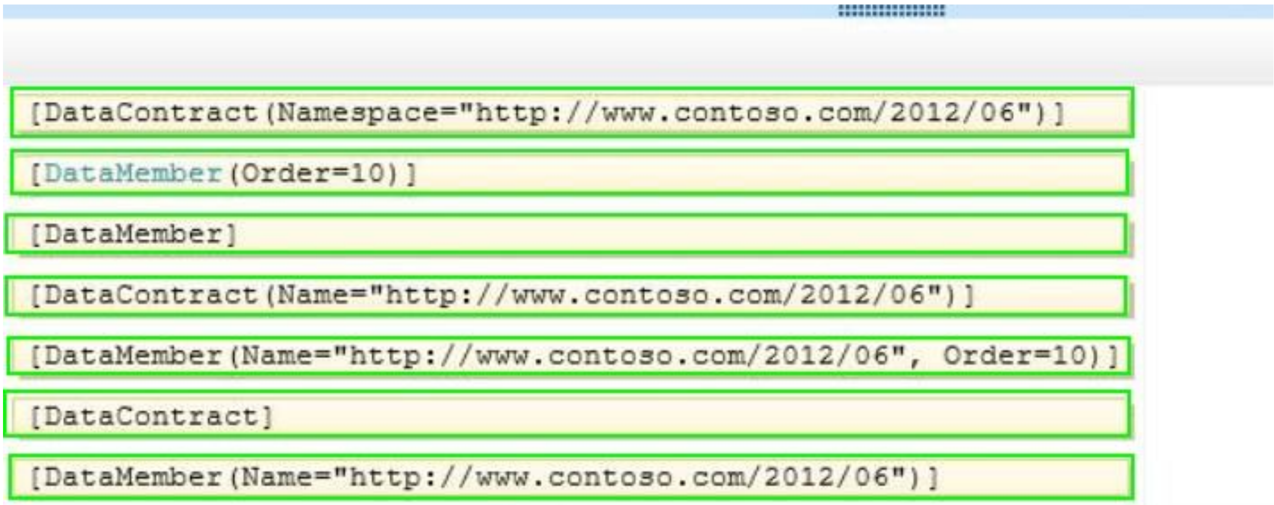
How should you complete the relevant code? (To answer, drag the appropriate attributes to the correct locations in the answer area-Each attribute may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.)



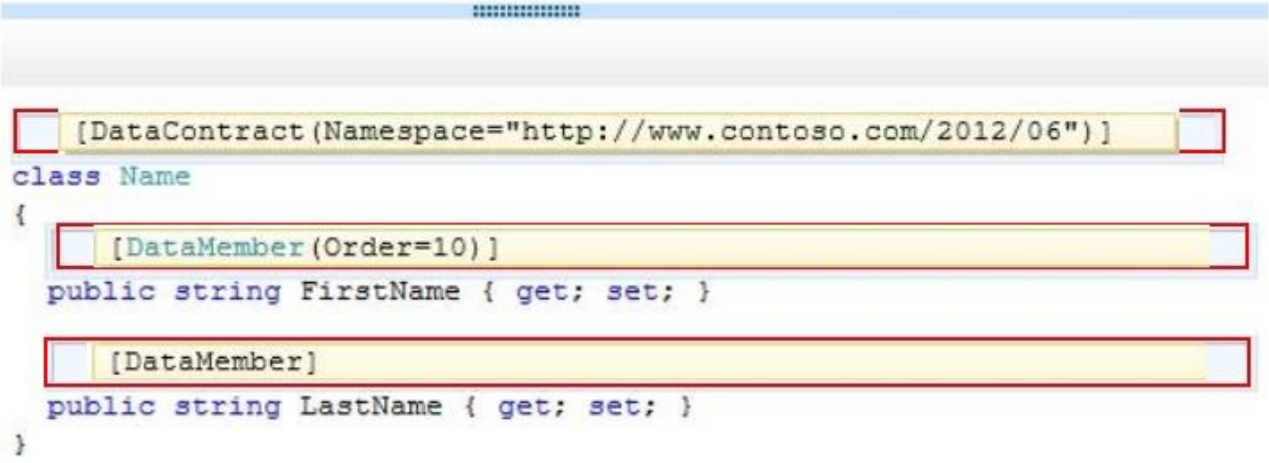
```
[DataContract(Namespace="http://www.contoso.com/2012/06")]
[DataMember(Order=10)]
[DataMember]
[DataContract(Name="http://www.contoso.com/2012/06")]
[DataMember(Name="http://www.contoso.com/2012/06", Order=10)]
[DataContract]
[DataMember(Name="http://www.contoso.com/2012/06")]
```

```
class Name
{

  public string FirstName { get; set; }

  public string LastName { get; set; }
}
```

Answer:



```
[DataContract(Namespace="http://www.contoso.com/2012/06")]
[DataMember(Order=10)]
[DataMember]
[DataContract(Name="http://www.contoso.com/2012/06")]
[DataMember(Name="http://www.contoso.com/2012/06", Order=10)]
[DataContract]
[DataMember(Name="http://www.contoso.com/2012/06")]
```

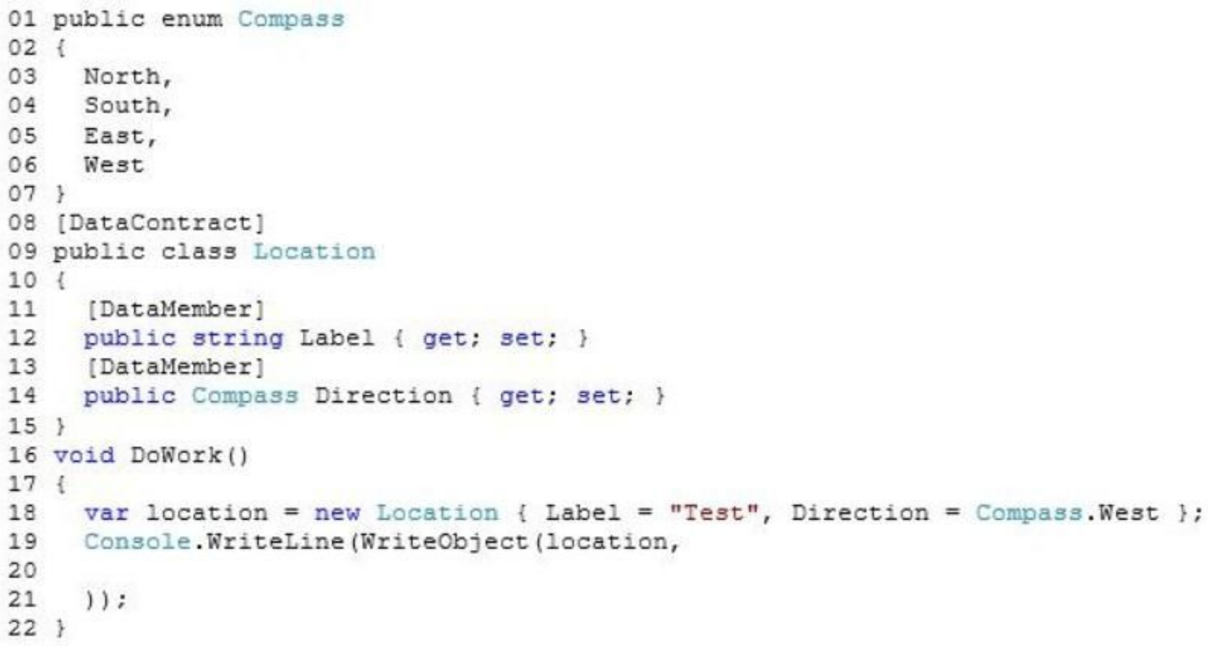


```
[DataContract(Namespace="http://www.contoso.com/2012/06")]
class Name
{
    [DataMember(Order=10)]
  public string FirstName { get; set; }

    [DataMember]
  public string LastName { get; set; }
}
```

Q 10

You are developing an application. The application converts a Location object to a string by using a method named WriteObject. The WriteObject() method accepts two parameters, a Location object and an XmlObjectSerializer object.

The application includes the following code. (Line numbers are included for reference only.)



```
01 public enum Compass
02 {
03   North,
04   South,
05   East,
06   West
07 }
08 [DataContract]
09 public class Location
10 {
11   [DataMember]
12   public string Label { get; set; }
13   [DataMember]
14   public Compass Direction { get; set; }
15 }
16 void DoWork()
17 {
18   var location = new Location { Label = "Test", Direction = Compass.West };
19   Console.WriteLine(WriteObject(location,
20
21   ));
22 }
```

You need to serialize the Location object as a JSON object.

Which code segment should you insert at line 20?

A. New DataContractSerializer(typeof(Location))
B. New XmlSerializer(typeof(Location))
C. New NetDataContractSenalizer()
D. New DataContractJsonSerializer(typeof(Location))

Answer: D

Explanation: The DataContractJsonSerializer class serializes objects to the JavaScript Object Notation (JSON) and deserializes JSON data to objects. Use the DataContractJsonSerializer class to serialize instances of a type into a JSON document and to deserialize a JSON document into an instance of a type.