



Guarantee All Exams 100% Pass Or Full Money Back

Vendor: Microsoft

Exam Code: 70-483

Exam Name: Microsoft Programming in C#

Questions and Answers No.: 151-160 (231Q&As)

- ☆ 100% Pass Guaranteed Or Full Money Back!
- ☆ Instant Download Access After Payment!
- ☆ One Year Free Updation!
- ☆ Well Formated: PDF,VCE,Exam Software!
- ☆ Multi-Platform capabilities - Windows, Laptop, Mac, Android, iPhone, iPod, iPad.
- ☆ Pass any exams at the FIRST try!

Get Latest&Actual 70-483 Exam Dumps from Braindump2go
<http://www.braindump2go.com/70-483.html>

QUESTION 151

Drag and Drop Question

You are developing a C# console application that outputs information to the screen.

The following code segments implement the two classes responsible for making calls to the Console object:

```
abstract class BaseLogger
{
    public virtual void Log(string message)
    {
        Console.WriteLine("Base: " + message);
    }
    public void LogCompleted()
    {
        Console.WriteLine("Completed");
    }
}

class Logger : BaseLogger
{
    public override void Log(string message)
    {
        Console.WriteLine(message);
    }
    public new void LogCompleted()
    {
        Console.WriteLine("Finished");
    }
}
```

When the application is run, the console output must be the following text:

- Log started
- Base: Log continuing
- Finished

You need to ensure that the application outputs the correct text.

Which four lines of code should you use in sequence? (To answer, move the appropriate classes from the list of classes to the answer area and arrange them in the correct order.)

logger.Log("Base: Log continuing");

((BaseLogger)logger).Log("Log continuing");

var logger = new BaseLogger();

((Logger)logger).LogCompleted();

logger.Log("Log started");

BaseLogger logger = new Logger();

logger.LogCompleted();

Answer:

<code>logger.Log("Base: Log continuing");</code>	<code>BaseLogger logger = new Logger();</code>
<code>((BaseLogger)logger).Log("Log continuing");</code>	<code>logger.Log("Log started");</code>
<code>var logger = new BaseLogger();</code>	<code>logger.Log("Base: Log continuing");</code>
<code>((Logger)logger).LogCompleted();</code>	<code>logger.LogCompleted();</code>
<code>logger.Log("Log started");</code>	
<code>BaseLogger logger = new Logger();</code>	
<code>logger.LogCompleted();</code>	

QUESTION 152

You are creating a class named Employee.

The class exposes a string property named EmployeeType.

The following code segment defines the Employee class. (Line numbers are included for reference only.)

The EmployeeType property value must meet the following requirements:

- The value must be accessed only by code within the Employee class or within a class derived from the Employee class.
- The value must be modified only by code within the Employee class.

You need to ensure that the implementation of the EmployeeType property meets the requirements.

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

```

01 public class Employee
02 {
03     internal string EmployeeType
04     {
05         get;
06         set;
07     }
08 }

```

- A. Replace line 03 with the following code segment:
public string EmployeeType
- B. Replace line 06 with the following code segment:
protected set;
- C. Replace line 05 with the following code segment:
private get;
- D. Replace line 05 with the following code segment:
protected get;
- E. Replace line 03 with the following code segment:
protected string EmployeeType
- F. Replace line 06 with the following code segment:
private set;

Answer: EF

QUESTION 153

You are developing an application that retrieves patient data from a web service. The application stores the JSON messages returned from the web service in a string variable named PatientAsJson. The variable is encoded as UTF-8.

The application includes a class named Patient that is defined by the following code:

```
public class Patient
{
    public bool IsActive { get; set; }
    public string Name { get; set; }
    public int Id { get; set; }
}
```

You need to populate the Patient class with the data returned from the web service.

Which code segment should you use?

- A. `DataContractJsonSerializer jsSerializer = new DataContractJsonSerializer(typeof(Patient));`
`using (MemoryStream stream = new MemoryStream(Encoding.UTF8.GetBytes(PatientAsJson)))`
`{`
 `Patient patientFromJson = (Patient)jsSerializer.ReadObject(stream);`
`}`
- B. `XmlSerializer xmlSerializer = new XmlSerializer(typeof(Patient));`
`using (MemoryStream stream = new MemoryStream(Encoding.UTF8.GetBytes(PatientAsJson)))`
`{`
 `Patient patientFromJson = (Patient)xmlSerializer.Deserialize(stream);`
`}`
- C. `DataContractJsonSerializer jsSerializer = new DataContractJsonSerializer(typeof(Patient));`
`using (MemoryStream stream = new MemoryStream(Encoding.UTF8.GetBytes(PatientAsJson)))`
`{`
 `Patient patientFromJson = new Patient();`
 `jsSerializer.WriteObject(stream, patientFromJson);`
`}`
- D. `IFormatter formatter = new BinaryFormatter();`
`Stream stream = new FileStream(PatientAsJson, FileMode.Open, FileAccess.Read, FileShare.Read);`
`Patient patientFromJson = (Patient)formatter.Deserialize(stream);`
`stream.Close();`

- A. Option A
B. Option B
C. Option C
D. Option D

Answer: A

QUESTION 154

Hotspot Question

You have an existing order processing system that accepts .xml files,
The following code shows an example of a properly formatted order in XML:

```
<Order OrderID="42">
  <Customer>Ben Smith</Customer>
  <CustomerID>206</CustomerID>
  <OrderDate>2013-04-19T09:13:14.7265994-05:00</OrderDate>
</Order>
```

You create the following class that will be serialized:

```
[DataContract()]
public class Order
{
    [DataMember()]
    public Int32 OrderID { get; set; }

    [DataMember(Name = "Customer")]
    public String CustomerName { get; set; }

    [DataMember()]
    private Int32 CustomerID { get; set; }

    public DateTime OrderDate { get; set; }
}
```

For each of the following properties, select Yes if the property is serialized according to the defined schema. Otherwise, select No.

	Yes	No
OrderID	<input type="radio"/>	<input type="radio"/>
OrderDate	<input type="radio"/>	<input type="radio"/>
CustomerName	<input type="radio"/>	<input type="radio"/>

Answer:

	Yes	No
OrderID	<input type="radio"/>	<input checked="" type="radio"/>
OrderDate	<input type="radio"/>	<input checked="" type="radio"/>
CustomerName	<input checked="" type="radio"/>	<input type="radio"/>

QUESTION 155

You are creating a console application named App1.

App1 will validate user input for order entries.

You are developing the following code segment (line numbers are included for reference only):

```
01 Console.WriteLine("Enter unit price: ");
02 string price = Console.ReadLine();
03
04 Console.WriteLine("Valid price");
05 else
06 Console.WriteLine("Invalid price")
```

You need to complete the code segment.

The solution must ensure that prices are positive and have two decimal places.

Which code should you insert at line 03?

- A. `if (!Regex.IsMatch(price, @"^(-)?\d+(\.\d\d)?$"))`
- B. `if (Regex.IsMatch(price, @"^(-)?\d+(\.\d\d)?$"))`
- C. `Regex reg = new Regex(@"^(-)?\d+(\.\d\d)?$");`
`if (reg.IsMatch(price))`
- D. `Regex reg = new Regex(@"^(-)?\d+(\.\d\d)?$");`
`if (reg.IsMatch(price))`

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Answer: B

Explanation:

- `Regex.IsMatch` Method (String, String)

Indicates whether the specified regular expression finds a match in the specified input string.

Syntax:

```
public static bool IsMatch(
    string input,
    string pattern
)
```

QUESTION 156

You are modifying an existing application that manages employee payroll.

The application includes a class named `PayrollProcessor`.

The `PayrollProcessor` class connects to a payroll database and processes batches of paychecks once a week.

You need to ensure that the `PayrollProcessor` class supports iteration and releases database connections after the batch processing completes.

Which two interfaces should you implement? (Each correct answer presents part of the complete solution. Choose two.)

- A. `IEquatable`
- B. `IEnumerable`

- C. IDisposable
- D. IComparable

Answer: BC

Explanation:

B: IEnumerable

C: IDisposable Interface

Exposes an enumerator, which supports a simple iteration over a non-generic collection.

Defines a method to release allocated resources.

The primary use of this interface is to release unmanaged resources.

QUESTION 157

You need to create a method that can be called by using a varying number of parameters. What should you use?

- A. method overloading
- B. interface
- C. named parameters
- D. lambda expressions

Answer: A

Explanation:

Member overloading means creating two or more members on the same type that differ only in the number or type of parameters but have the same name. Overloading is one of the most important techniques for improving usability, productivity, and readability of reusable libraries. Overloading on the number of parameters makes it possible to provide simpler versions of constructors and methods. Overloading on the parameter type makes it possible to use the same member name for members performing identical operations on a selected set of different types.

QUESTION 158

You write the following method (line numbers are included for reference only):

```
01 public static List<string> TestIfWebSite(string url)
02 {
03     const string pattern = @"http://(www\.)?([^\.\.]+)\.com";
04     List<string> result = new List<string>();
05
06     MatchCollection myMatches = Regex.Matches(url, pattern);
07     ...
08     return result;
09 }
```

You need to ensure that the method extracts a list of URLs that match the following pattern:

@http://(www\.)?([^\.\.]+)\.com;

Which code should you insert at line 07?

- A. `result = (List<string>) myMatches.SyncRoot;`
- B. `result = (from System.Text.RegularExpressions.Match m in myMatches
where m.Value.Contains(pattern)
select m.Value).ToList<string>();`
- C. `foreach (Match currentMatch in myMatches)
result.Add(currentMatch.Groups.ToString());`
- D. `foreach (Match currentMatch in myMatches)
result.Add(currentMatch.Value);`

- A. Option A
B. Option B
C. Option C
D. Option D

Answer: D

Explanation:

- MatchCollection

Represents the set of successful matches found by iteratively applying a regular expression pattern to the input string.

The collection is immutable (read-only) and has no public constructor.

The `Regex.Matches` method returns a `MatchCollection` object.

- `List<T>.Add` Method

Adds an object to the end of the `List<T>`.

Incorrect:

Not A: `ICollection.SyncRoot` Property

For collections whose underlying store is not publicly available, the expected implementation is to return the current instance. Note that the pointer to the current instance might not be sufficient for collections that wrap other collections; those should return the underlying collection's `SyncRoot` property.

QUESTION 159

You are developing an application by using C#.

The application includes a method named `SendMessage`.

The `SendMessage()` method requires a string input.

You need to replace "Hello" with "Goodbye" in the parameter that is passed to the `SendMessage()` method.

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

- ☐ A. `var message = "Hello World";
SendMessage(message.Replace("Goodbye", "Hello"));`
- ☐ B. `var message = "Hello World";
SendMessage(message.Replace("Hello", "Goodbye"));`
- ☐ C. `var message = "Hello World";
message = message.Replace("Hello", "Goodbye");
SendMessage(message);`
- ☐ D. `var message = "Hello World";
message.Replace("Goodbye", "Hello");
SendMessage(message);`

- A. Option A
B. Option B
C. Option C
D. Option D

Answer: BC

Explanation:

- The first parameter should be Hello.

- String.Replace Method (String, String)

Returns a new string in which all occurrences of a specified string in the current instance are replaced with another specified string.

This method does not modify the value of the current instance.

Instead, it returns a new string in which all occurrences of oldValue are replaced by newValue.

QUESTION 160

You are developing an application by using C#.

The application includes an object that performs a long running process.

You need to ensure that the garbage collector does not release the object's resources until the process completes.

Which garbage collector method should you use?

- A. `WaitForFullGCCComplete()`
B. `SuppressFinalize()`
C. `WaitForFullGCApproach()`
D. `WaitForPendingFinalizers()`

Answer: B