Vendor: Microsoft

Exam Code: 70-483

Exam Name: Microsoft Programming in C#

Questions and Answers No.: 21-30 (231Q&As)

QUESTION 21
Hotspot Question
You are implementing a library method that accepts a character parameter and returns a string.
If the lookup succeeds, the method must return the corresponding string value.
If the lookup fails, the method must return the value "invalid choice."
You need to implement the lookup algorithm.
How should you complete the relevant code? (To answer, select the correct keyword in each drop-down list in the answer area.)

Work Area

```
public string GetResponse(char letter)
{
   string response;
   [switch ▾] (letter)
   {
      [case ▾] 'a':
         response = "animal";
         break;
      [case ▾] 'm':
         response = "mineral";
         break;
      [case ▾]:
         response = "invalid choice";
         break;
   }
   return response;
}
```

Answer:

Work Area

```
public string GetResponse(char letter)
{
   string response;
   [switch] (letter)
   {
      [case] 'a':
         response = "animal";
         break;
      [case] 'm':
         response = "mineral";
         break;
      [default]:
         response = "invalid choice";
         break;
   }
   return response;
}
```

QUESTION 22
You use the Task.Run() method to launch a long-running data processing operation.
The data processing operation often fails in times of heavy network congestion.
If the data processing operation fails, a second operation must clean up any results of the first operation.
You need to ensure that the second operation is invoked only if the data processing operation throws an unhandled exception.
What should you do?

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

QUESTION 23
You are modifying an application that processes leases.
The following code defines the Lease class. (Line numbers are included for reference only.)

```
01 public class Lease
02 {
03
04   private int _term;
05   private const int MaximumTerm = 5;
06   private const decimal Rate = 0.034m;
07   public int Term
08   {
09     get
10     {
11       return _term;
12     }
13     set
14     {
15       if (value <= MaximumTerm)
16       {
17         _term = value;
18       }
19       else
20       {
21
22       }
23     }
24   }
25 }
26 public delegate void MaximumTermReachedHandler(object source, EventArgs e);
```

Leases are restricted to a maximum term of 5 years.
The application must send a notification message if a lease request exceeds 5 years.
You need to implement the notification mechanism.
Which two actions should you perform? (Each correct answer presents part of the solution.
Choose two.)

A. Insert the following code segment at line 03:

```
public event MaximumTermReachedHandler OnMaximumTermReached;
```

B. Insert the following code segment at line 21:

```
if (OnMaximumTermReached != null)
{
  OnMaximumTermReached(this, new EventArgs());
}
```

C. Insert the following code segment at line 21:

```
value = MaximumTerm;
```

D. Insert the following code segment at line 03:

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

E. Insert the following code segment at line 03:

```
private string MaximumTermReachedEvent;
```

F. Insert the following code segment at line 21:

```
value = 4;
```

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

Answer: AB

QUESTION 24
You are developing an application that uses structured exception handling.
The application includes a class named ExceptionLogger.
The ExceptionLogger class implements a method named LogException by using the following code segment:

```
public static void LogException(Exception ex)
```

You have the following requirements:

```
- Log all exceptions by using the LogException() method of the
ExceptionLogger class.
- Rethrow the original exception, including the entire exception stack.
```

You need to meet the requirements.
Which code segment should you use?

A.
```
catch (Exception ex)
{
  ExceptionLogger.LogException(ex);
  throw;
}
```

B.
```
catch (Exception ex)
{
  ExceptionLogger.LogException(ex);
  throw ex;
}
```

C.
```
catch
{
  ExceptionLogger.LogException(new Exception());
  throw;
}
```

D.
```
catch
{
  var ex = new Exception();
  throw ex;
}
```

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

Answer: A

QUESTION 25
You are developing an application that includes a class named UserTracker.
The application includes the following code segment. (Line numbers are included for reference only.)

```
01 public delegate void AddUserCallback(int i);
02 public class UserTracker
03 {
04   List<User> users = new List<User>();
05   public void AddUser(string name, AddUserCallback callback)
06   {
07     users.Add(new User(name));
08     callback(users.Count);
09   }
10 }
11
12 public class Runner
13 {
14
15   UserTracker tracker = new UserTracker();
16   public void Add(string name)
17   {
18
19   }
20 }
```

You need to add a user to the UserTracker instance.
What should you do?

A. Insert the following code segment at line 14:

```
private static void PrintUserCount(int i)
{
  ...
}
```

Insert the following code segment at line 18:

```
AddUserCallback callback = PrintUserCount;
```

B. Insert the following code segment at line 11:

```
delegate void AddUserDelegate(UserTracker userTracker);
```

Insert the following code segment at line 18:

```
AddUserDelegate addDelegate = (userTracker) =>
  {
    ...
  };
addDelegate(tracker);
```

C. Insert the following code segment at line 11:

```
delegate void AddUserDelegate(string name, AddUserCallback callback);
```

Insert the following code segment at line 18:

```
AddUserDelegate adder = (i, callback) =>
  {
    ...
  };
```

D. Insert the following code segment at line 18:

```
tracker.AddUser(name, delegate(int i)
  {
    ...
  });
```

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

Answer: D

QUESTION 26
Drag and Drop Question
You develop an application that displays information from log files.
When a user opens a log file by using the application, the application throws an exception and closes.
The application must preserve the original stack trace information when an exception occurs.
You need to implement the method that reads the log files.
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.)

```
using (StringReader sr = new StringReader("log.txt"))
using (StreamReader sr = new StreamReader("log.txt"))
throw new FileNotFoundException();
throw;
```

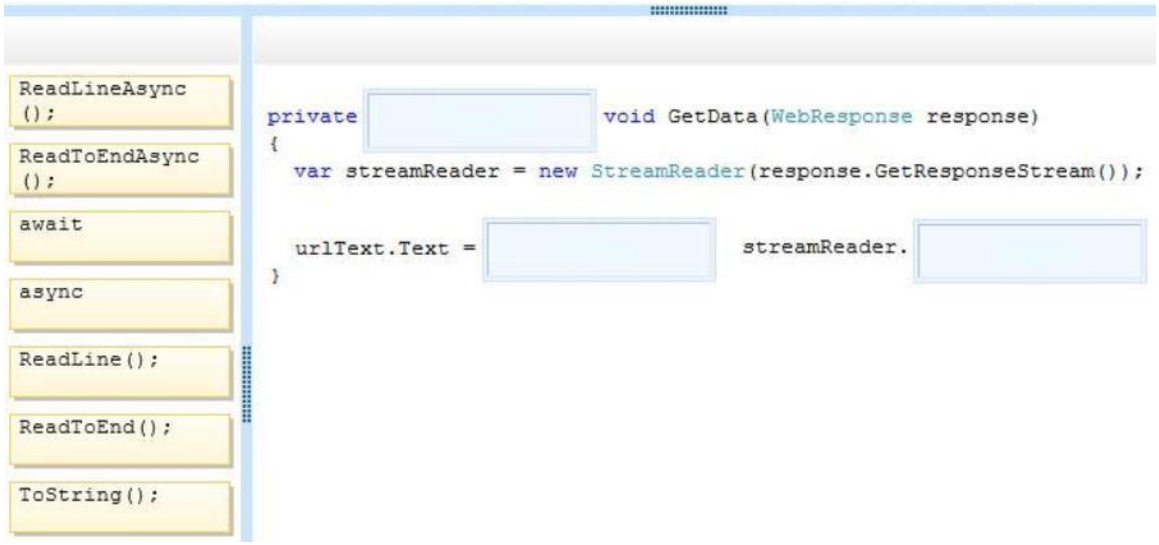
```
{
    try
    {
        string line;
        while ((line = sr.ReadLine()) != null)
        {
            Console.WriteLine(line);
        }
    }
    catch (FileNotFoundException e)
    {
        Console.Write(e.ToString());

    }
}
```

Answer:

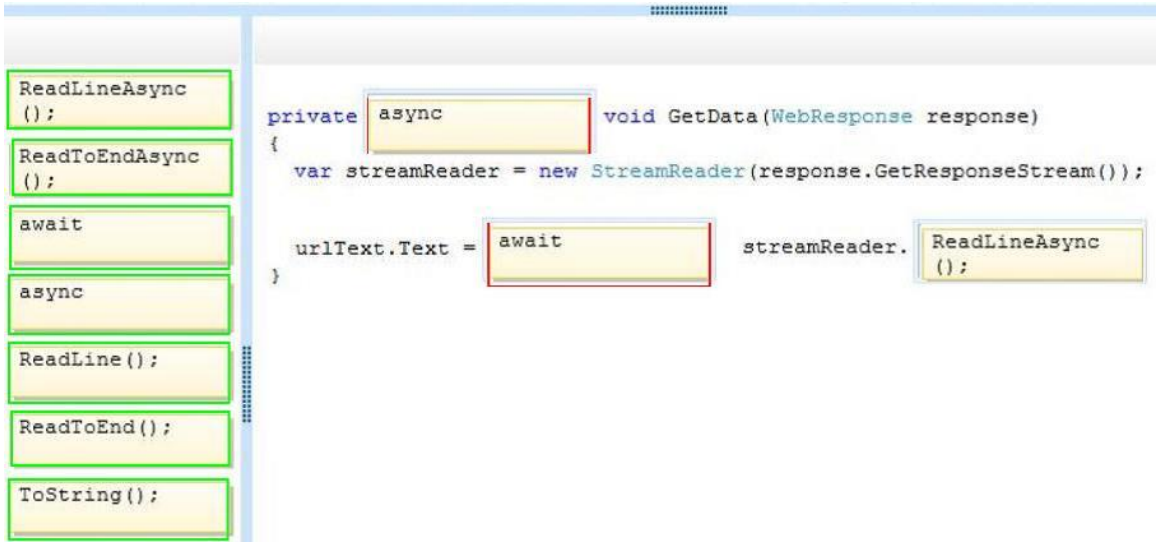
```
using (StringReader sr = new StringReader("log.txt"))
using (StreamReader sr = new StreamReader("log.txt"))
throw new FileNotFoundException();
throw;
```

```
using (StreamReader sr = new StreamReader("log.txt"))
{
    try
    {
        string line;
        while ((line = sr.ReadLine()) != null)
        {
            Console.WriteLine(line);
        }
    }
    catch (FileNotFoundException e)
    {
        Console.Write(e.ToString());
        throw;
    }
}
```

QUESTION 27
Drag and Drop Question
You are developing an application that will include a method named GetData.
The GetData() method will retrieve several lines of data from a web service by using a System.IO.StreamReader object.
You have the following requirements:
- The GetData() method must return a string value that contains the first line of the response from the web service.
- The application must remain responsive while the GetData() method runs.
You need to implement the GetData() method.
How should you complete the relevant code? (To answer, drag the appropriate objects to the correct locations in the answer area. Each object 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.)



Answer:



QUESTION 28

You are adding a public method named UpdateScore to a public class named ScoreCard.
The code region that updates the score field must meet the following requirements:

- It must be accessed by only one thread at a time.
- It must not be vulnerable to a deadlock situation.

You need to implement the UpdateScore() method.
What should you do?

A. Place the code region inside the following lock statement:

```
lock (this)
{
  ...
}
```

B. Add a private object named **lockObject** to the **ScoreCard** class. Place the code region inside the following lock statement:

```
lock (lockObject)
{
  ...
}
```

C. Apply the following attribute to the **UpdateScore()** method signature:

```
[MethodImpl(MethodImplOptions.Synchronized)]
```

D. Add a public static object named **lockObject** to the **ScoreCard** class. Place the code region inside the following lock statement:

```
lock (typeof(ScoreCard))
{
  ...
}
```

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

Answer: B

QUESTION 29
Drag and Drop Question
You are developing an application that includes a class named Kiosk.
The Kiosk class includes a static property named Catalog.
The Kiosk class is defined by the following code segment. (Line numbers are included for reference only.)

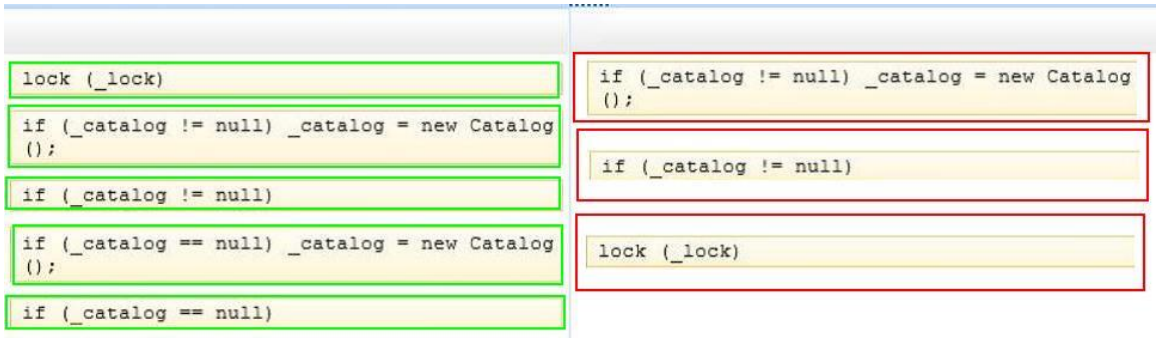
```
01 public class Kiosk
02 {
03   static Catalog _catalog = null;
04   static object _lock = new object();
05   public static Catalog Catalog
06   {
07     get
08     {
09
10       return _catalog;
11     }
12   }
13 }
```

You have the following requirements:
- Initialize the _catalog field to a Catalog instance.
- Initialize the _catalog field only once.
- Ensure that the application code acquires a lock only when the _catalog object must be instantiated.

You need to meet the requirements.

Which three code segments should you insert in sequence at line 09? (To answer, move the appropriate code segments from the list of code segments to the answer area and arrange them in the correct order.)

Answer:



QUESTION 30

Drag and Drop Question

You are developing an application that implements a set of custom exception types.

You declare the custom exception types by using the following code segments:

```
public class ContosoException : System.Exception { ... }
public class ContosoDbException : ContosoException { ... }
public class ContosoValidationException : ContosoException { ... }
```

The application includes a function named DoWork that throws .NET Framework exceptions and custom exceptions.

The application contains only the following logging methods:

```
static void Log(Exception ex) { ... }
static void Log(ContosoException ex) { ... }
static void Log(ContosoValidationException ex) { ... }
```

The application must meet the following requirements:

- When AdventureWorksValidationException exceptions are caught, log the information by using the static void Log (AdventureWorksValidationException ex) method.
- When AdventureWorksDbException or other AdventureWorksException exceptions are caught, log the information by using the static void I oq(AdventureWorksException ex) method.

You need to meet the requirements.

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.)

(AdventureWorksValidationException ex)
(AdventureWorksException ex)
(Exception ex)
(ContosoDbException ex)

```
try
{
  DoWork();
}
catch
{
  Log(ex);
}
catch
{
  Log(ex);
}
catch
{
  Log(ex);
}
```

Answer:

(AdventureWorksValidationException ex)
(AdventureWorksException ex)
(Exception ex)
(ContosoDbException ex)

```
try
{
  DoWork();
}
catch (AdventureWorksValidationException ex)
{
  Log(ex);
}
catch (AdventureWorksException ex)
{
  Log(ex);
}
catch (Exception ex)
{
  Log(ex);
}
```