70-483 Q&As

Programming in C#

Pass Microsoft 70-483 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.geekcert.com/70-483.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Microsoft Official Exam Center

QUESTION 1

You are implementing a new method named ProcessData. The ProcessData() method calls a third-party component that performs a long-running operation. The third-party component uses the IAsyncResult pattern to signal completion of the long- running operation.

You need to ensure that the calling code handles the long-running operation as a System.Threading.Tasks.Task object.

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

A. Call the component by using the TaskFactory.FromAsync() method.

B. Create a TaskCompletionSource object.

C. Apply the async modifier to the method signature.

D. Apply the following attribute to the method signature: [MethodImpl(MethodImplOptions.Synchronized)]

Correct Answer: AB

Explanation: A: TaskFactory.FromAsync Method

Creates a Task that represents a pair of begin and end methods that conform to the Asynchronous Programming Model pattern. Overloaded.

Example:

TaskFactory.FromAsync Method (IAsyncResult, Action) Creates a Task that executes an end method action when a specified IAsyncResult completes.

B: In many scenarios, it is useful to enable a Task to represent an external asynchronous operation. TaskCompletionSource is provided for this purpose. It enables the creation of a task that can be handed out to consumers, and those consumers can use the members of the task as they would any other. However, unlike most tasks, the state of a task created by a TaskCompletionSource is controlled explicitly by the methods on TaskCompletionSource. This enables the completion of the external asynchronous operation to be propagated to the underlying Task. The separation also ensures that consumers are not able to transition the state without access to the corresponding TaskCompletionSource.

Note:

* System.Threading.Tasks.Task Represents an asynchronous operation.

QUESTION 2

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. WaitForFullGCComplete()

B. WaitForFullGCApproach()

C. KeepAlive()

D. WaitForPendingFinalizers()

Correct Answer: C

Explanation: The GC.KeepAlive method references the specified object, which makes it ineligible for garbage collection from the start of the current routine to the point where this method is called.

The purpose of the KeepAlive method is to ensure the existence of a reference to an object that is at risk of being prematurely reclaimed by the garbage collector. The KeepAlive method performs no operation and produces no side effects

other than extending the lifetime of the object passed in as a parameter.

Reference: GC.KeepAlive Method (Object)

https://msdn.microsoft.com/en-us/library/system.gc.keepalive(v=vs.110).aspx

QUESTION 3

You are creating a console application named App1.

App1 retrieves data from the Internet by using JavaScript Object Notation (JSON).

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

```
01 public bool ValidateJson(string json, Dictionary<string, object> result)
02 {
03
04   try
05   {
06     result = serializer.DeserializeDictionary<string, object>>(json);
07     return true;
08   }
09   catch
10   {
11     return false;
12   }
13 }
```


You need to ensure that the code validates the JSON string. Which code should you insert at line 03?

A.
```
var serializer = new DataContractSerializer();
```

B.
```
DataContractSerializer serializer = new DataContractSerializer();
```

C.
```
var serializer = new XmlSerializer();
```

D.
```
var serializer = new JavaScriptSerializer();
```


A. Option A

B. Option B

C. Option C

D. Option D

Correct Answer: D

Explanation: The JavaScriptSerializer Class Provides serialization and deserialization functionality for AJAX-enabled applications.

The JavaScriptSerializer class is used internally by the asynchronous communication layer to serialize and deserialize the data that is passed between the browser and the Web server. You cannot access that instance of the serializer.

However, this class exposes a public API. Therefore, you can use the class when you want to work with JavaScript Object Notation (JSON) in managed code.

QUESTION 4

DRAG DROP

You have an application that accesses a Microsoft SQL Server database.

The database contains a stored procedure named Proc1. Proc1 accesses several rows of data across multiple tables.

You need to ensure that after Proc1 executes, the database is left in a consistent state. While Proc1 executes, no other operation can modify data already read or changed by Proc1. (Develop the solution by selecting and ordering the required

code snippets.

You may not need all of the code snippets.)

Select and Place:

```
SqlTransaction transaction = connection.Begi
nTransaction
(System.Data.IsolationLevel.RepeatableRead);

SqlTransaction transaction = connection.Begi
nTransaction
(System.Data.IsolationLevel.ReadUncommitted)
;

} finally {

command.Dispose();
connection.Dispose();
}

try {
connection.Open();
command.ExecuteNonQuery();

TransactionScope transaction = new Transacti
onScope();

SqlConnection connection = new SqlConnection
(connectionString);
SqlCommand command = new SqlCommand
("proc1", connection);

} catch {

transaction.Rollback();

transaction.Commit();
```


Correct Answer:

QUESTION 5

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

Correct Answer: A

Once an exception is thrown, part of the information it carries is the stack trace. The stack trace is a list of the method call hierarchy that starts with the method that throws the exception and ends with the method that catches the exception. If an exception is re- thrown by specifying the exception in the throw statement, the stack trace is restarted at the current method and the list of method calls between the original method that threw the exception and the current method is lost. To keep the original stack trace information with the exception, use the throw statement without specifying the exception.

Reference: http://msdn.microsoft.com/en-us/library/ms182363(v=vs.110).aspx