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.pass4lead.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 need to write a console application that meets the following requirements:

If the application is compiled in Debug mode, the console output must display Entering debug mode.

If the application is compiled in Release mode, the console output must display Entering release mode.

Which code should you use?

A.
```
#define DEBUG
  Console.WriteLine("Entering debug mode");
#define RELEASE
  Console.WriteLine("Entering release mode");
```

B.
```
if(System.Reflection.Assembly.GetExecutingAssembly().IsDefined
 (typeof(System.Diagnostics.Debugger), false))
  Console.WriteLine("Entering debug mode");
else
  Console.WriteLine("Entering release mode");
```

C.
```
#region DEBUG
  Console.WriteLine("Entering debug mode");
#endregion
#region RELEASE
  Console.WriteLine("Entering release mode");
#endregion
```

D.
```
#if (DEBUG)
  Console.WriteLine("Entering debug mode");
#elif (RELEASE)
  Console.WriteLine("Entering release mode ");
#endif
```


A. Option A

B. Option B

C. Option C

D. Option D

Correct Answer: D

Explanation: #elif lets you create a compound conditional directive. The #elif expression will be evaluated if neither the preceding #if (C# Reference) nor any preceding, optional, #elif directive expressions evaluate to true. If a #elif expression evaluates to true, the compiler evaluates all the code between the #elif and the next conditional directive.

For example: #define VC7 //... #if debug Console.Writeline("Debug build"); #elif VC7 Console.Writeline("Visual Studio 7"); #endif

Incorrect: Not B:

*

System.Reflection.Assembly.GetExecutingAssembly Method Gets the assembly that contains the code that is currently executing.* Assembly.IsDefined Method Indicates whether or not a specified attribute has been applied to the assembly.

*

System.Dignostics.Debugger Class Enables communication with a debugger.

Property: IsAttached

Gets a value that indicates whether a debugger is attached to the process.

QUESTION 2

You are modifying an existing banking application.

The application includes an Account class and a Customer class. The following code segment defines the classes.

```
class Account
{
  public Account(decimal balance, int term, decimal rate)
  {
    Term = term;
    Balance = balance;
    Rate = rate;
  }
  public decimal Balance { get; set; }
  public decimal Rate { get; set; }
  public int Term { get; set; }
}

class Customer
{
  public Customer(string firstName, string lastName, Collection<Account> accounts)
  {
    FirstName = firstName;
    LastName = lastName;
    AccountCollection = accounts;
  }
  public string FirstName { get; set; }
  public string LastName { get; set; }
  public Collection<Account> AccountCollection { get; set; }
}
```


You populate a collection named customerCollection with Customer and Account objects by using the following code segment:

```
Collection<Customer> customerCollection = new Collection<Customer>();
Collection<Account> customerAccounts = new Collection<Account>();
customerAccounts.Add(new Account(1000m, 2, 0.025m));
customerAccounts.Add(new Account(3000m, 4, 0.045m));
customerAccounts.Add(new Account(5000m, 6, 0.045m));
customerCollection.Add(new Customer("David", "Jones", customerAccounts));
```

You create a largeCustomerAccounts collection to store the Account objects by using the following code segment:

Collection largeCustomerAccounts = new Collection ();

All accounts with a Balance value greater than or equal to 1,000,000 must be tracked.

You need to populate the largeCustomerAccounts collection with Account objects.

Which code segment should you use?

A.
```
foreach (Customer customer in customerCollection)
{
  foreach (Account account in customer.AccountCollection)
  {
    if (account.Balance >= 1000000m)
    {
      customer.AccountCollection.Add(account);
    }
  }
}
```

B.
```
foreach (Account customer in customerCollection)
{
  foreach (Account account in largeCustomerAccounts)
  {
    if (account.Balance >= 1000000m)
    {
      largeCustomerAccounts.Add(account);
    }
  }
}
```

C.
```
foreach (Customer customer in customerCollection)
{
  foreach (Account account in customer.AccountCollection)
  {
    if (account.Balance >= 1000000m)
    {
      largeCustomerAccounts.Add(account);
    }
  }
}
```

D.
```
foreach (Account account in largeCustomerAccounts)
{
  foreach (Customer customer in customerCollection)
  {
    if (account.Balance >= 1000000m)
    {
      customer.AccountCollection.Add(account);
    }
  }
}
```

A. Option A

B. Option B

C. Option C

D. Option D

Correct Answer: C

QUESTION 3

HOTSPOT

You have the following code.

```
public class Order
  {
      public int OrderId { get; set;}
      public DateTime { get; set;}
      public Order(int orderId, DateTime OrderDate)
      {
          OrderId = OrderId;
          OrderDate = OrderDate;
      }
  }
public class OrderDetails : Order
 {
      public string ProductName { get; set; }
      public OrderDetails(string productName, int orderId,DateTime orderDate)
        : base(OrderId, OrderDate)
      {
          ProductName = ProductName;
      }
 }
```


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

Hot Area:

Statement	Yes	No
The OrderId property is inherited by OrderDetails.	○	○
A new property named ProductName is added to the Order constructor.	○	○
OrderId and OrderDate are required parameters when you create OrderDetails objects.	○	○

Correct Answer:

Statement	Yes	No
The OrderId property is inherited by OrderDetails.	●	○
A new property named ProductName is added to the Order constructor.	○	●
OrderId and OrderDate are required parameters when you create OrderDetails objects.	●	○

QUESTION 4

You are developing an application by using C#. The application includes the following code segment. (Line numbers are included for reference only.)

```
01 public interface IDataContainer
02 {
03   string Data { get; set; }
04 }
05 void DoWork(object obj)
06 {
07
08   if (dataContainer != null)
09   {
10     Console.WriteLine(dataContainer.Data);
11   }
12 }
```


The DoWork() method must throw an InvalidCastException exception if the obj object is not of type IDataContainer when accessing the Data property. You need to meet the requirements. Which code segment should you insert at line 07?

A. var dataContainer = (IDataContainer) obj;

B. var dataContainer = obj as IDataContainer;

C. var dataContainer = obj is IDataContainer;

D. dynamic dataContainer = obj;

Correct Answer: A

direct cast. If object is not of the given type, an InvalidCastException is thrown.

Incorrect:

Not B: If obj is not of the given type, result is null. Not C: If obj is not of a given type, result is false. Not D: This simply check the variable during runtime. It will not throw an exception.

Reference: http://msdn.microsoft.com/en-us/library/ms173105.aspx

QUESTION 5

You are developing a method named CreateCounters that will create performance counters for an application. The method includes the following code. (Line numbers are included for reference only.)

```
01 void CreateCounters()
02 {
03   if (!PerformanceCounterCategory.Exists("Contoso"))
04   {
05     var counters = new CounterCreationDataCollection();
06     var ccdCounter1 = new CounterCreationData
07     {
08       CounterName = "Counter1",
09       CounterType = PerformanceCounterType.AverageTimer32
11     };
12     counters.Add(ccdCounter1);
13     var ccdCounter2 = new CounterCreationData
14     {
15       CounterName = "Counter2",
16
17     };
18     counters.Add(ccdCounter2);
19     PerformanceCounterCategory.Create("Contoso", "Help string",
20      PerformanceCounterCategoryType.MultiInstance, counters);
21   }
22 }
```


You need to ensure that Counter2 is available for use in Windows Performance Monitor (PerfMon). Which code segment should you insert at line 16?

A. CounterType = PerformanceCounterType.RawBase

B. CounterType = PerformanceCounterType.AverageBase

C. CounterType = PerformanceCounterType.SampleBase

D. CounterType = PerformanceCounterType.CounterMultiBase

Correct Answer: B

Explanation: Note AverageTimer32 on line 09. The Base counter type AverageBase has the Parent (composite) counter types AverageTimer32, AverageCount64. Reference:

http://msdn.microsoft.com/en-us/library/system.diagnostics.performancecountertype.aspx

To Read the Whole Q&As, please purchase the Complete Version from Our website.

Try our product !

100% Guaranteed Success
100% Money Back Guarantee
365 Days Free Update
Instant Download After Purchase
24x7 Customer Support
Average 99.9% Success Rate
More than 800,000 Satisfied Customers Worldwide
Multi-Platform capabilities - Windows, Mac, Android, iPhone, iPod, iPad, Kindle

We provide exam PDF and VCE of Cisco, Microsoft, IBM, CompTIA, Oracle and other IT Certifications.
You can view Vendor list of All Certification Exams offered:

https://www.pass4lead.com/allproducts

Need Help

Please provide as much detail as possible so we can best assist you.
To update a previously submitted ticket: