



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.: 101-110 (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 101

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<Account> largeCustomerAccounts = new Collection<Account>();
```

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

Answer: C

QUESTION 102

Drag and Drop Question

You have a method that will evaluate a parameter of type Int32 named Status.

You need to ensure that the method meets the following requirements:

- If Status is set to Active, the method must return 1.
- If Status is set to Inactive, the method must return 0.
- If Status is any other value, the method must return -1.

What should you do? (To answer, drag the appropriate statement to the correct location in the answer area. Each statement 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.)

break;

case "Active":

case "Inactive"

default:

goto default;

return

```
Int32 returnStatus = Int32.MinValue;

switch (status) {
    Statement
    returnStatus = 1;
    Statement
    returnStatus = 0;
    Statement
    returnStatus = -1;
    Statement
}

return returnStatus;
```

Answer:

break;

case "Active":

case "Inactive"

default:

goto default;

return

```
Int32 returnStatus = Int32.MinValue;

switch (status) {
    case "Active":
        returnStatus = 1;
        break;
    case "Inactive"
        returnStatus = 0;
        break;
    default:
        returnStatus = -1;
        break;
}

return returnStatus;
```

QUESTION 103

You are developing an application that uses several objects.

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

```
01 private bool IsNull(object obj)
02 {
03
04     return false;
05 }
```

You need to evaluate whether an object is null.
Which code segment should you insert at line 03?

- A.

```
if (null = obj)
{
    return true;
}
```
- B.

```
if (null == obj)
{
    return true;
}
```
- C.

```
if (null)
{
    return true;
}
```
- D.

```
if (!obj)
{
    return true;
}
```

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

Answer: B

Explanation:

Use the == operator to compare values and in this case also use the null literal.

QUESTION 104

You are creating a class named Loan.

The Loan class must meet the following requirements:

- Include a member that represents the rate for a Loan instance.
- Allow external code to assign a value to the rate member.
- Restrict the range of values that can be assigned to the rate member.

You need to implement the rate member to meet the requirements.

In which form should you implement the rate member?

- A. public static property
- B. public property
- C. public static field
- D. protected field

Answer: B

QUESTION 105

You are developing an application that will parse a large amount of text.
You need to parse the text into separate lines and minimize memory use while processing data.
Which object type should you use?

- A. DataContractSerializer
- B. StringBuilder
- C. StringReader
- D. JsonSerializer

Answer: C

QUESTION 106

You are developing an assembly.
You plan to sign the assembly when the assembly is developed.
You need to reserve space in the assembly for the signature.
What should you do?

- A. Run the Assembly Linker tool from the Windows Software Development Kit (Windows SDK).
- B. Run the Strong Name tool from the Windows Software Development Kit (Windows SDK).
- C. Add the AssemblySignatureKeyAttribute attribute to the assembly.
- D. Add the AssemblyDelaySignAttribute attribute to the assembly.

Answer: D

QUESTION 107

You are developing an application that includes a class named Employee and a generic list of employees.

The following code segment declares the list of employees:

```
List<Employee> employeesList = new List<Employee>();
```

You populate the employeesList object with several hundred Employee objects.

The application must display the data for five Employee objects at a time.

You need to create a method that will return the correct number of Employee objects.

Which code segment should you use?

- A. `public static IEnumerable<int> Page(IEnumerable<int> source, int page, int pageSize)`
`{`
 `return source.Take((pageSize - 1) * page).Skip(pageSize);`
`}`
- B. `public static IEnumerable<TSource> Page<TSource>(this IEnumerable<TSource> source, int page, int pageSize)`
`{`
 `return source.Skip((page - 1) * pageSize).Take(pageSize);`
`}`
- C. `public static IEnumerable<int> Page(IEnumerable<int> source, int page, int pageSize)`
`{`
 `return source.Skip((pageSize - 1) * page).Take(pageSize);`
`}`
- D. `public static IEnumerable<TSource> Page<TSource>(this IEnumerable<TSource> source, int page, int pageSize)`
`{`
 `return source.Take((page - 1) * pageSize).Skip(pageSize);`
`}`

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

Answer: B

QUESTION 108

You are implementing a new method named `ProcessData`.

The `ProcessData()` method calls a third-party component that performs a long-running operation to retrieve stock information from a web service.

The third-party component uses the `IAsyncResult` pattern to signal completion of the long-running operation so that the UI can be updated with the new values.

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

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

- A. Create a `TaskCompletionSource<T>` object.
B. Call the component by using the `TaskFactory.FromAsync()` method.
C. Apply the following attribute to the `ProcessData()` method signature:
`[MethodImpl(MethodImplOptions.Synchronized)]`
D. Apply the `async` modifier to the `ProcessData()` method signature.

Answer: AB

Explanation

A: In many scenarios, it is useful to enable a `Task<TResult>` to represent an external asynchronous operation. `TaskCompletionSource<TResult>` 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`.

B: `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<IAsyncResult>)

Creates a Task that executes an end method action when a specified IAsyncResult completes.

Note:

- System.Threading.Tasks.Task

Represents an asynchronous operation.

QUESTION 109

You are evaluating a method that calculates loan interest.

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

When the loanTerm value is 5 and the loanAmount value is 4500, the loanRate must be set to 6.5 percent.

You need to adjust the loanRate value to meet the requirements.

What should you do?

```
01 private static decimal CalculateInterest(decimal loanAmount, int loanTerm)
02 {
03     decimal interestAmount;
04     decimal loanRate;
05     if (loanTerm > 0 && loanTerm < 5 && loanAmount < 5000m)
06     {
07         loanRate = 0.045m;
08     }
09     else if (loanTerm > 5 && loanAmount > 5000m)
10     {
11         loanRate = 0.085m;
12     }
13     else
14     {
15         loanRate = 0.055m;
16     }
17     interestAmount = loanAmount * loanRate * loanTerm;
18     return interestAmount;
19 }
```

- A. Replace line 15 with the following code segment:
loanRate = 0.065m;
- B. Replace line 07 with the following code segment:
loanRate = 0.065m;
- C. Replace line 17 with the following code segment:
interestAmount = loanAmount * 0.065m * loanTerm;
- D. Replace line 04 with the following code segment:
decimal loanRate = 0.065m;

Answer: A

QUESTION 110

You are creating a class library that will be used in a web application.

You need to ensure that the class library assembly is strongly named.

What should you do?

- A. use the csc.exe /target:Library option when building the application.
- B. use the AL.exe command-line tool.
- C. use the aspnet_regiis.exe command-line tool.
- D. use the EdmGen.exe command-line tool.

Answer: B

Explanation:

The Windows Software Development Kit (SDK) provides several ways to sign an assembly with a strong name:

- Using the Assembly Linker (Al.exe) provided by the Windows SDK.
- Using assembly attributes to insert the strong name information in your code.

You can use either the AssemblyKeyFileAttribute or the AssemblyKeyNameAttribute, depending on where the key file to be used is located.

- Using compiler options such /keyfile or /delaysign in C# and Visual Basic, or the /KEYFILE or /DELAYSIGN linker option in C++. (For information on delay signing, see Delay Signing an Assembly.)

Note:

- A strong name consists of the assembly's identity--its simple text name, version number, and culture information (if provided)--plus a public key and a digital signature. It is generated from an assembly file (the file that contains the assembly manifest, which in turn contains the names and hashes of all the files that make up the assembly), using the corresponding private key.

Microsoft® Visual Studio® .NET and other development tools provided in the .NET Framework SDK can assign strong names to an assembly.

Assemblies with the same strong name are expected to be identical.