Vendor: Microsoft

Exam Code: 70-483

Exam Name: Microsoft Programming in C#

Questions and Answers No.: 111-120 (231Q&As)

QUESTION 111
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 (obj = null)
{
  return true;
}
```

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

C.
```
if (obj ==  0)
{
  return true;
}
```

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

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

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

QUESTION 112
You are developing an application.
The application contains the following code:
When you compile the code, you receive the following syntax error message:
"A previous catch clause already catches all exceptions of this or a super type ('System.Exception')."
You need to ensure that the code can be compiled.
What should you do?

```
class Program
{
  static void ProcessOrders (string orderRefNumber)
  {
    if (orderRefNumber == null)
    {
      throw new ArgumentNullException();
    }
    ...
  }

  static void Main()
  {
    try
    {
      string orderRefNumber = null;
      ProcessOrders(orderRefNumber);
    }
    catch (ArgumentNullException e)
    {
      Console.WriteLine("{0} An exception caught.", e);
    }

    catch (Exception e)
    {
      Console.WriteLine("{0} An exception caught.", e);
    }
  }
}
```

A. Catch the ArgumentException exception instead of the ArgumentNullException exception.
B. Throw a new exception in the second catch block.
C. Catch the ArgumentNullException exception first.
D. Re-throw the exception caught by the second catch block.

Answer: A

QUESTION 113
You plan to store passwords in a Windows Azure SQL Database database.
You need to ensure that the passwords are stored in the database by using a hash algorithm,
Which cryptographic algorithm should you use?

A. ECDSA
B. RSA-768
C. AES-256
D. SHA-256

Answer: D

QUESTION 114
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 3 and the loanAmount value is 9750, the loanRate must be set to 8.25 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 04 with the following code segment:
`decimal loanRate = 0.0325m;`
B. Replace line 17 with the following code segment:
`interestAmount = loanAmount * 0.0825m * loanTerm;`
C. Replace line 15 with the following code segment:
`loanRate = 0.0825m;`
D. Replace line 07 with the following code segment:
`loanRate = 0.0825m;`

Answer: C

QUESTION 115
You are developing code for an application that retrieves information about Microsoft .NET Framework assemblies.
The following code segment is part of the application (line numbers are included for reference only):
You need to insert code at line 04.
The code must load the assembly.
Once the assembly is loaded, the code must be able to read the assembly metadata, but the code must be denied access from executing code from the assembly.
Which code segment should you insert at line 04?

```
01 public void ViewMetadata(string filePath)
02 {
03   var bytes = File.ReadAllBytes(filePath);
04
05   ...
06 }
```

A. `Assembly.ReflectionOnlyLoadFrom(bytes);`

B. `Assembly.ReflectionOniyLoad(bytes);`
C. `Assembly.Load(bytes);`
D. `Assembly.LoadFrom(bytes);`

Answer: C

QUESTION 116
Hotspot Question
You are reviewing the following code:

```
[System.FlagsAttribute()]
public enum Group
{
  Users = 1,
  Supervisors = 2,
  Managers = 4,
  Administrators = 8
}
public class User
{
  public Group UserGroup { get; set; }
}
```

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

	Yes	No
A user can be a member of more than one of the groups.	○	○
If the user belongs to only the Administrators group, the following code will return a value of true: `user.UserGroup == Group.Administrators`	○	○
If the user belongs to only the Supervisors group, the following code will return a value of true: `user.UserGroup < Group.Administrators`	○	○

Answer:

	Yes	No
A user can be a member of more than one of the groups.	○	○
If the user belongs to only the Administrators group, the following code will return a value of true: user.UserGroup == Group.Administrators	○	○
If the user belongs to only the Supervisors group, the following code will return a value of true: user.UserGroup < Group.Administrators	○	○

QUESTION 117
You are developing a method named GetHash that will return a hash value for a file.
The method includes the following code. (Line numbers are included for reference only.)

```
01 public byte[] GetHash(string filename, string algorithmType)
02 {
03    var hasher = HashAlgorithm.Create(algorithmType);
04    var fileBytes = System.IO.File.ReadAllBytes(filename);
05
06 }
```

You need to return the cryptographic hash of the bytes contained in the fileBytes variable.
Which code segment should you insert at line 05?

A.
```
var outputBuffer = new byte[fileBytes.Length];
hasher.TransformBlock(fileBytes, 0, fileBytes.Length, outputBuffer, 0);
hasher.TransformFinalBlock(fileBytes, fileBytes.Length - 1, fileBytes.Length);
return outputBuffer;
```

B.
```
hasher.ComputeHash(fileBytes);
return hasher.GetHashCode();
```

C.
```
var outputBuffer = new byte[fileBytes.Length];
hasher.TransformBlock(fileBytes, 0, fileBytes.Length, outputBuffer, 0);
return outputBuffer;
```

D.
```
hasher.ComputeHash(fileBytes);
return hasher.Hash;
```

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

Answer: A

QUESTION 118
You are debugging a 64-bit C# application.
Users report System.OutOfMemoryException exceptions.
The system is attempting to use arrays larger than 2 GB in size.
You need to ensure that the application can use arrays larger than 2 GB.
What should you do?

A. Add the /3GB switch to the boot.ini file for the operating system.
B. set the IMAGE_FILE_LARGE_ADDRESS_AWARE flag in the image header for the application executable file.
C. set the value of the gcAllowVeryLargeObjects property to true in the application configuration file.
D. Set the value of the user-mode virtual address space setting for the operating system to MAX.

Answer: C

QUESTION 119
Drag and Drop Question
You are developing a class named Temperature.
You need to ensure that collections of Temperature objects are sortable.
How should you complete the relevant code segment? (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.)

```
public class Temperature : IComparable
public class Temperature : IComparer
CompareTo
Equals
this.Fahrenheit.CompareTo(otherTemperature.Fahrenheit);
otherTemperature.Fahrenheit.CompareTo(this.Fahrenheit);
```

```
{
    public double Fahrenheit { get; set; }
    public int
        (object obj)
    {
        if (obj == null) return 1;
        var otherTemperature = obj as Temperature;
        if(otherTemperature != null)
            return
        throw new ArgumentException("Object is not a Temperature");
    }
}
```

Answer:

```
public class Temperature : IComparable
public class Temperature : IComparer
CompareTo
Equals
this.Fahrenheit.CompareTo(otherTemperature.Fahrenheit);
otherTemperature.Fahrenheit.CompareTo(this.Fahrenheit);
```

```
public class Temperature : IComparable
{
    public double Fahrenheit { get; set; }
    public int CompareTo
        (object obj)
    {
        if (obj == null) return 1;
        var otherTemperature = obj as Temperature;
        if(otherTemperature != null)
            return this.Fahrenheit.CompareTo(otherTemperature.Fahrenheit);
        throw new ArgumentException("Object is not a Temperature");
    }
}
```

QUESTION 120
Drag and Drop Question
You are adding a method to an existing application.
The method uses an integer named statusCode as an input parameter and returns the status code as a string.
The method must meet the following requirements:
- Return "Error" if the statusCode is 0.
- Return "Success" if the statusCode is 1.
- Return "Unauthorized" if the statusCode is any value other than 0 or 1.

You need to implement the method to meet the requirements.
How should you complete the relevant code? (To answer, drag the appropriate statements to the correct locations 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.)

default
switch
break
case

```
string statusText;
        (statusCode)
{
        0:
            statusText = "Error";
            ;
        1:
            statusText = "Success";
            ;
        :
            statusText = "Unauthorized";
            ;
}
return statusText;
```

Answer:

default
switch
break
case

```
string statusText;
switch (statusCode)
{
    case 0:
            statusText = "Error";
        break ;
    case 1:
            statusText = "Success";
        break ;
    default :
            statusText = "Unauthorized";
        break ;
}
return statusText;
```