70-483

Microsoft
Programming in C#

The 70-483 practice exam is written and formatted by Certified Senior IT Professionals working in today's prospering companies and data centers all over the world! The 70-483 Practice Test covers all the exam topics and objectives and will prepare you for success quickly and efficiently. The 70-483 exam is very challenging, but with our 70-483 questions and answers practice exam, you can feel confident in obtaining your success on the 70-483 exam on your FIRST TRY!

Microsoft 70-483 Exam Features

- Detailed questions and answers for 70-483 exam
- Try a demo before buying any Microsoft exam
- 70-483 questions and answers, updated regularly
- Verified 70-483 answers by Experts and bear almost 100% accuracy
- 70-483 tested and verified before publishing
- 70-483 exam questions with exhibits
- 70-483 same questions as real exam with multiple choice options

Acquiring Microsoft certifications are becoming a huge task in the field of I.T. More over these exams like 70-483 exam are now continuously updating and accepting this challenge is itself a task. This 70-483 test is an important part of Microsoft certifications. We have the resources to prepare you for this. The 70-483 exam is essential and core part of Microsoft certifications and once you clear the exam you will be able to solve the real life problems yourself.Want to take advantage of the Real 70-483 Test and save time and money while developing your skills to pass your Microsoft 70-483 Exam? Let us help you climb that ladder of success and pass your 70-483 now!

QUESTION: 1

You are developing an application that includes a class named Order. The application will store a collection of Order objects. The collection must meet the following requirements:
-Use strongly typed members.
-Process Order objects in first-in-first-out order.
-Store values for each Order object.
-Use zero-based indices.
You need to use a collection type that meets the requirements.
Which collection type should you use?

A. Queue <T>
B. SortedList
C. LinkedList <T>
D. HashTable
E. Array <T>

Answer: A

QUESTION: 2

You are developing an application. The application calls a method that returns an array of integers named employeeIds. You define an integer variable named employeeIdToRemove and assign a value to it. You declare an array named filteredEmployeeIds. You have the following requirements:
-Remove duplicate integers from the employeeIds array.
-Sort the array in order from the highest value to the lowest value.
-Remove the integer value stored in the employeeIdToRemove variable from the employeeIds array.You need to create a LINQ query to meet the requirements.
Which code segment should you use?

```
C A.   int[] filteredEmployeeIds = employeeIds.Where(value => value !=
         employeeIdToRemove).OrderBy(x => x).ToArray();

C B.   int[] filteredEmployeeIds = employeeIds.Where(value => value !=
         employeeIdToRemove).OrderByDescending(x => x).ToArray();

C C.   int[] filteredEmployeeIds = employeeIds.Distinct().Where(value => value !=
         employeeIdToRemove).OrderByDescending(x => x).ToArray();

C D.   int[] filteredEmployeeIds = employeeIds.Distinct().OrderByDescending(x => x).ToArray();
```

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

Answer: C

QUESTION: 3

You are developing an application that includes the following code segment. (Line numbers are included for reference only.)

```
01 class Animal
02 {
03    public string Color { get; set; }
04    public string Name { get; set; }
05 }
06 private static IEnumerable<Animal> GetAnimals(string sqlConnectionString)
07 {
08    var animals = new List<Animal>();
09    SqlConnection sqlConnection = new SqlConnection(sqlConnectionString);
10    using (sqlConnection)
11    {
12       SqlCommand sqlCommand = new SqlCommand("SELECT Name, ColorName FROM Animals", sqlConnection);
13
14       using (SqlDataReader sqlDataReader = sqlCommand.ExecuteReader())
15       {
16
17          {
18             var animal = new Animal();
19             animal.Name = (string)sqlDataReader["Name"];
20             animal.Color = (string)sqlDataReader["ColorName"];
21             animals.Add(animal);
22          }
23       }
24    }
25    return customers;
26 }
```

The GetAnimals() method must meet the following requirements:
- Connect to a Microsoft SQL Server database.
- Create Animal objects and populate them with data from the database.
- Return a sequence of populated Animal objects.

You need to meet the requirements.

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

A. Insert the following code segment at line 16:
while (sqlDataReader.NextResult())
B. Insert the following code segment at line 13:
sqlConnection.BeginTransaction();
C. Insert the following code segment at line 13:
sqlConnection.Open();
D. Insert the following code segment at line 16:
while (sqlDataReader.Read())
E. insert the following code segment at line 16:
while (sqlDataReader.GetValues())

Answer: C, D

QUESTION: 4

DRAG DROP

You are developing a custom collection named LoanCollection for a class named Loan class. You need to ensure that you can process each Loan object in the LoanCollection collection by using a foreach loop. 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.)

```
: IComparable

: IEnumerable

: IDisposable

public IEnumerator GetEnumerator()

public int CompareTo(object obj)

public void Dispose()

_loanCollection[0].Amount++;

return obj == null ? 1 : _loanCollection.Length;

return _loanCollection.GetEnumerator();
```

```
public class LoanCollection
{
    private readonly Loan[] _loanCollection;
    public LoanCollection(Loan[] loanArray)
    {
        _loanCollection = new Loan[loanArray.Length];

        for (int i = 0; i < loanArray.Length; i++)
        {
            _loanCollection[i] = loanArray[i];
        }
    }



    {

    }
}
```

Answer:

```
: IComparable
```

```
: IEnumerable
```

```
: IDisposable
```

```
public IEnumerator GetEnumerator()
```

```
public int CompareTo(object obj)
```

```
public void Dispose()
```

```
loanCollection[0].Amount++;
```

```
return obj == null ? 1 : _loanCollection.Length;
```

```
return _loanCollection.GetEnumerator();
```
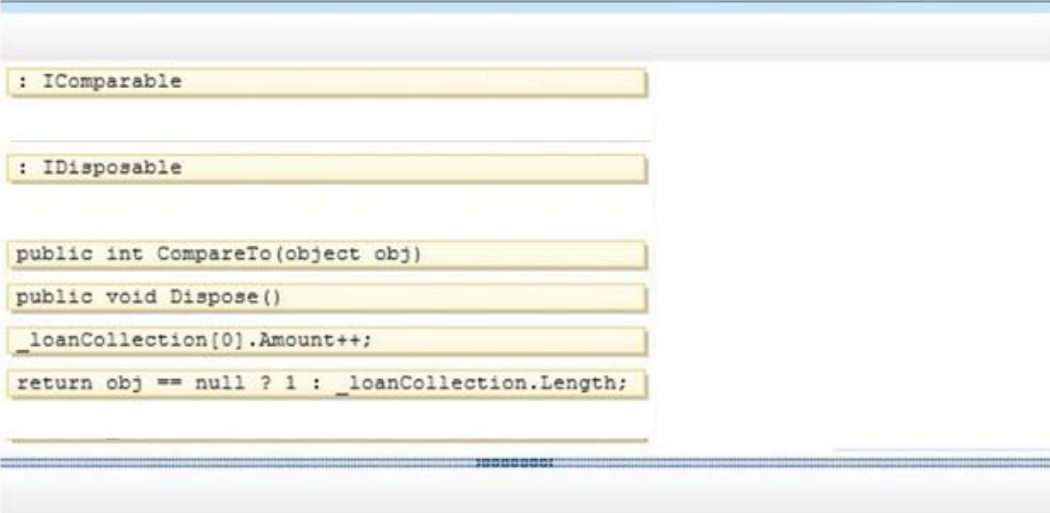
```
public class LoanCollection : IEnumerable
{
    private readonly Loan[] _loanCollection;
    public LoanCollection(Loan[] loanArray)
    {
        _loanCollection = new Loan[loanArray.Length];

        for (int i = 0; i < loanArray.Length; i++)
        {
            _loanCollection[i] = loanArray[i];
        }
    }

    public IEnumerator GetEnumerator()
    {
        return _loanCollection.GetEnumerator();
    }
}
```

```
: IComparable


: IDisposable


public int CompareTo(object obj)

public void Dispose()

_loanCollection[0].Amount++;

return obj == null ? 1 : _loanCollection.Length;
```

:::::::::

```
public class LoanCollection  : IEnumerable
{
    private readonly Loan[] _loanCollection;
    public LoanCollection(Loan[] loanArray)
    {
        _loanCollection = new Loan[loanArray.Length];

        for (int i = 0; i < loanArray.Length; i++)
        {
            _loanCollection[i] = loanArray[i];
        }
    }

    public IEnumerator GetEnumerator()
    {
        return _loanCollection.GetEnumerator();
    }
}
```

QUESTION: 5

You are developing an application that uses the Microsoft ADO.NET Entity Framework to retrieve order information from a Microsoft SQL Server database. The application includes the following code. (Line numbers are included for reference only.)

```
01 public DateTime? OrderDate;
02 IQueryable<Order> LookupOrdersForYear(int year)
03 {
04    using (var context = new NorthwindEntities())
05    {
06      var orders =
07       from order in context.Orders
08
09        select order;
10      return orders.ToList().AsQueryable();
11    }
12 }
```

The application must meet the following requirements:
- Return only orders that have an OrderDate value other than null.
- Return only orders that were placed in the year specified in the OrderDate property or in a later year.
You need to ensure that the application meets the requirements.
Which code segment should you insert at line 08?

A. Where order.OrderDate.Value != null && order.OrderDate.Value.Year > = year
B. Where order.OrderDate.Value = = null && order.OrderDate.Value.Year = = year
C. Where order.OrderDate.HasValue && order.OrderDate.Value.Year = = year
D. Where order.OrderDate.Value.Year = = year

Answer: C

QUESTION: 6
DRAG DROP
You are developing an application by using C#. The application includes an array of decimal values named loanAmounts. You are developing a LINQ query to return the values from the array. The query must return decimal values that are evenly divisible by two. The values must be sorted from the lowest value to the highest value. You need to ensure that the query correctly returns the decimal values. 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.)

```
join
from
group
ascending
descending
where
orderby
select
```

```
decimal[] loanAmounts = { 303m, 1000m, 85579m, 501.51m, 603m
   1200m, 400m, 22m };
IEnumerable<decimal> loanQuery =
            amount in loanAmounts
            amount % 2 == 0
            amount [        ]
            amount;
```

Answer:

```
join
from
group
ascending
descending
where
orderby
select
```

```
decimal[] loanAmounts = { 303m, 1000m, 85579m, 501.51m, 603m
   1200m, 400m, 22m };
IEnumerable<decimal> loanQuery =
   from      amount in loanAmounts
   where     amount % 2 == 0
   orderby   amount  ascending
   select    amount;
```

Box 1: from Box 2: where
Box 3: orderby Box 4: ascending
Box 5: select
Note: In a query expression, the orderby clause causes the returned sequence or
subsequence (group) to be sorted in either ascending or descending order.
Examples:
// Query for ascending sort. IEnumerable<string> sortAscendingQuery = from
fruit in fruits
orderby fruit //"ascending" is default select fruit;
// Query for descending sort. IEnumerable<string> sortDescendingQuery = from
w in fruits
orderby w descending select w;

QUESTION: 7
You are developing an application. The application includes a method named
ReadFile that reads data from a file.
The ReadFile() method must meet the following requirements:

Pass4sure Certification Exam Features;

- Pass4sure offers over **2500** Certification exams for professionals.
- More than **98,800** Satisfied Customers Worldwide.
- Average **99.8%** Success Rate.
- Over **120** Global Certification Vendors Covered.
- Services of Professional & Certified Experts available via support.
- Free 90 days updates to match real exam scenarios.
- Instant Download Access! No Setup required.
- Price as low as $19, which is 80% more cost effective than others.
- Verified answers researched by industry experts.
- Study Material **updated** on regular basis.
- Questions / Answers are downloadable in **PDF** format.
- Mobile Device Supported (Android, iPhone, iPod, iPad)
- **No authorization** code required to open exam.
- **Portable** anywhere.
- *Guaranteed Success*.
- **Fast**, helpful support 24x7.

View list of All certification exams offered;
 http://www.ipass4sure.com/allexams.asp

View list of All Study Guides (SG);
http://www.ipass4sure.com/study-guides.asp

View list of All Audio Exams (AE);
http://www.ipass4sure.com/audio-exams.asp

Download Any Certication Exam DEMO.
http://www.ipass4sure.com/samples.asp

To purchase Full version of exam click below;
http://www.ipass4sure.com/allexams.asp

3COM	CompTIA	Filemaker	IBM	LPI	OMG	Sun
ADOBE	ComputerAssociates	Fortinet	IISFA	McAfee	Oracle	Sybase
APC	CWNP	Foundry	Intel	McData	PMI	Symantec
Apple	DELL	Fujitsu	ISACA	Microsoft	Polycom	TeraData
BEA	ECCouncil	GuidanceSoftware	ISC2	Mile2	RedHat	TIA
BICSI	EMC	HDI	ISEB	NetworkAppliance	Sair	Tibco
CheckPoint	Enterasys	Hitachi	ISM	Network-General	SASInstitute	TruSecure
Cisco	ExamExpress	HP	Juniper	Nokia	SCP	Veritas
Citrix	Exin	Huawei	Legato	Nortel	See-Beyond	Vmware
CIW	ExtremeNetworks	Hyperion	Lotus	Novell	Google	

and many others.. See complete list Here