Q.1) Consider the following code snippet and then answer the given question.

Public delegate void changeEvent();

public event changeEvent changed;

if (changed!=null) changed();

Which of the following statements would you write

1. public delegate void changeEvent();
2. public event changeEvent changed;
3. List.changed+=new changeEvent(ListChanged);
4. List.changed-=new changeEvent(ListChanged);

Q.2) Randy wants to use a type that provides call-back capabilities for events.   
 Given the scenario above, which one of the following types should he use?

1. Delegate 2. Context 3. Transaction 4. Pointer 5. Attribute

Q.3) Statement A : Delegates are instances that give a name to a method signature

Statement B: Delegates are not Type-Safe

Statement C: An instance of a given delegate can refer to any instance or statuc method on

any obj

1. Statement A is true, B and C are false 2. Statement A and C are true, B is false

3. Statement A and B are true, C is false 4. Statement B and C are true, A is false

Q.4) Which one of the following exceptions is thrown when trying to change the value of a read-

only column?

1. System.Data.ReadOnlyException
2. System.Data.Common.ReadOnlyException
3. System.Data.SqlClient.ReadOnlyException
4. System.Data.LockedException
5. System.Data.FixedException

Q.5) Which one of the following objects do you use to catch error information from SQL Server?

1. System.Data.SqlClient.SqlError

2. System.Data.SqlClient.Error

3. System.Data.SqlClient.ErrorInformation

4. System.Data.SqlClient.Exception

5. System.Data.SqlClient.SqlException

Q.6) Which of the following statements are true?  
 Statement A : Multicast delegates invoke more than one methods.  
 Statement B : Singlecast delegates invoke exactly one method

1. Statement A is true 2. Statement B is true

3. Both statements are true 4. Both Statements are false

Q.7) fill up the following demo snippet with appropriate option :  
 using System;  
 public class SamplesDelegate {  
  
 // Declares a delegate for a method that takes in an int and returns a String.  
 public delegate String myMethodDelegate( int myInt );  
  
 // D

1. myMethodDelegate myD2 = new myMethodDelegate( mySampleClass.mySignMethod );

2. myMethodDelegate myD2 = new myMethodDelegate(mySC.mySignMethod );

3. myMethodDelegate myD2 = new myMethodDelegate(mySignMethod );

4. myMethodDelegate myD2 = new Delegate( mySampleClass.mySignMethod );

Q.8) Which of the following statements are true :  
 Statement A : When a delegate references a static method, the delegate stores a reference to the

method's entry point.

Statement B : A delegate can reference a method only if the signature of the method exactl

1. Statement A is True.
2. Statement B is True.
3. Both are True
4. Both are False.

Q.9) //Sample code for C# Exception tutorial using try , catch  
  
 // try catch exception  
 int zero = 0;  
 try  
 {  
 int div = 100/zero;  
 }  
 catch(DivideByZeroException)  
 {  
 Console.WriteLine("Division by zero exception passed");  
 1. This code in runtime throws a DivideByZeroException and writes some message through

the console.

}

2. This code in runtime throws a DivideByZeroException to Console.

3. This code gets executed successfully.

4. None of the above

Q.10) Which of the following code will be used to create a C# .Net Custom Exception class as

TextException class that inherits from System.Exception class of .NET class library. Actually it

does nothing, but there is an additional class MailValidator. It has Te

1. //Sample code for C# .Net Exception tutorial - validates an email address  
   public class TextException : Exception  
   {  
   public TextException() : base()<br>  
   {  
   }  
   public TextException(string message) : base(message)  
   {  
   }  
   }  
   public class MailValidator  
   {  
   Mail
2. //Sample code for C# .Net Exception tutorial - validates an email address  
   public class TextException : Exception  
   {  
   public TextException() : base()<br>  
   {  
   }  
   public TextException(string message) : base(message)  
   {  
   }  
   }  
   public class MailValidator  
   {  
   Mail
3. //Sample code for C# .Net Exception tutorial - validates an email address  
   public class TextException : BaseException  
   {  
   public TextException() : base()<br>  
   {  
   }  
   public TextException(string message) : base(message)  
   {  
   }  
   }  
   public class MailValidator  
   {
4. //Sample code for C# .Net Exception tutorial - validates an email address  
   public class TextException : Exception  
   {  
   public TextException() : base()<br>  
   {  
   }  
   public TextException(string message) : base(message)  
   {  
   }  
   }  
   public class MailValidator  
   {  
     
   Mail

Q.11) Statement A :If an application handles exceptions that occur during the execution of a block of

application code, the code must be placed within a try statement.

Statement B: Application code that handles exceptions thrown by a try block is placed within

1. Statement A is True 2. Statement B is True

3. Both Statements are True 4. Both Statements are false

Q.12) categories of exceptions exist under the base class Exceptionis/are: \_\_\_\_\_\_\_\_\_\_\_-

1. SystemException 2. ApplicationException

3. SystemException and ApplicationException. 4. [InnerException,](ms-help://MS.MSDNQTR.2003FEB.1033/cpref/html/frlrfsystemexceptionclassinnerexceptiontopic.htm)

Q.13) If an exception occurs during destructor execution,and that exception is not caught, then \_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. the destructor of the base class (if any) is called.If there is no base class (as in the case of

the object type) or if there is no base class destructor, then the exception is discarded.

2. the execution of that destructor is terminated and the destructor of the base class (if any)

is called.

3. the execution of that destructor is terminated and the destructor of the base class (if any)

is called.If there is no base class (as in the case of the object type) or if there is no base

class destructor, then the exception is discarded.

4. the exception is discarded.

Q.14) snippet :  
 using System;  
 class Test  
 {  
 static void F() {  
 try {  
 G();  
 }  
 catch (Exception e) {  
 Console.WriteLine("Exception in F: " + e.Message);  
 e = new Exception("F");  
 throw; // re-throw

1. a throw statement with no expression can be used to re-throw the exception that was

caught by the catch block.

2. a throw statement with no expression can not be used to re-throw the exception that was

caught by the catch block.

3. a throws statement with no expression can be used to re-throw the exception that was

caught by the catch block.

4. throw statement with no expression can be used to throw the exception that was caught

by the catch block.

Q.15) The process that takes place from the point of the exception being thrown to the point of

transferring control to a suitable exception handler is known as \_\_\_\_\_\_\_\_\_\_\_\_

1. exception propagation. 2. exception Handling

3. Throwing Exception 4. Finalizer