Section 1: Dockers

##### Q.**Identify true statements related to Docker : 1:Docker provides tooling and a platform to manage the lifecycle of your containers. 2:The Docker platform is the only container platform to build, secure and manage  the widest array of applications from development to production both on premises  and in the clouds.**

**1)**. Both are false

**2)**. Statement 1 is true Statement 2 is false

**3)**. **Statement 1 is false Statement 2 is true**

**4)**. **Both are true**

**Solution** :  
option [4] is correct

**Attempted** :  
option [3] is attempted

##### Q.**Complete the Image : https://icompass.fs.capgemini.com/ckeditorimages/1520412450406_Practice-Docker-Q10.jpg**

**1)**. **REST API**

**2)**. **Services**

**3)**. OS

**4)**. Server

**Solution** :  
option [1] is correct

**Attempted** :  
option [2] is attempted

##### Q.**\_\_\_\_\_\_\_\_\_\_ is used for building Docker images and creating Docker containers.**

**1)**. Image

**2)**. Container

**3)**. **Docker Engine**

**4)**. Server

**Solution** :  
option [3] is correct

**Attempted** :  
option [3] is attempted

##### Q.**Mohan has choseen SignalR implementation in.NET Framework.What type of container  he should use?**

**1)**. **Windows**

**2)**. Unix

**3)**. Mac

**4)**. Linux

**Solution** :  
option [1] is correct

**Attempted** :  
option [1] is attempted

##### Q.**Docker is configured to look for images on \_\_\_\_\_\_\_\_.**

**1)**. Containers

**2)**. Engine

**3)**. **Docker Hub**

**4)**. Docker Cloud

**Solution** :  
option [3] is correct

**Attempted** :  
option [3] is attempted

##### Q.**You need to start containers fast and want a small footprint per container to achieve better  density or more containers per hardware unit in order to lower your costs which containers  you will use ?**

**1)**. **Windows**

**2)**. **Linux**

**3)**. Mac

**4)**. **Unix**

**Solution** :  
option [1,2] are correct

**Attempted** :  
option [4] is attempted

##### Q.**Identify true statements related to Docker : 1:You can create, start, stop, move, or delete a container using the Docker API or CLI. 2:The Docker client (docker) is the primary way that many Docker users interact with Docker.**

**1)**. Both are false

**2)**. **Statement 1 is true Statement 2 is false**

**3)**. Statement 1 is false Statement 2 is true

**4)**. **Both are true**

**Solution** :  
option [4] is correct

**Attempted** :  
option [2] is attempted

##### Q.**\_\_\_\_\_\_\_\_&\_\_\_\_\_\_\_\_ are supported implementations for building server-side containerized  Docker applications with .NET.**

**1)**. **.NET CORE**

**2)**. **.NET Framework**

**3)**. Linux

**4)**. Unix

**Solution** :  
option [1,2] are correct

**Attempted** :  
option [1,2] are attempted

##### Q.**Your application is a Windows Server legacy app ("brown-field") migration to containers. Which container you should use?**

**1)**. **Windows**

**2)**. Mac

**3)**. **Linux**

**4)**. Unix

**Solution** :  
option [1] is correct

**Attempted** :  
option [3] is attempted

##### Q.**\_\_\_\_\_\_\_\_\_&\_\_\_\_\_\_\_\_\_ are public registries that anyone can use.**

**1)**. **Docker Hub**

**2)**. **Docker Cloud**

**3)**. **Docker Engine**

**4)**. Docker Images

**Solution** :  
option [1,2] are correct

**Attempted** :  
option [1,3] are attempted

#### Section 2: Developer Tools

##### Q.**Which of the following class is used by client application to request the logger instances?**

**1)**. **LogManager**

**2)**. Log

**3)**. Logger

**4)**. **Appender**

**Solution** :  
option [1] is correct

**Attempted** :  
option [4] is attempted

##### Q.**Rajesh has developed his ecommerce application now he needs a source analysis tool for his application. Which application you would recommend Rajesh?**

**1)**. Use Fxcop

**2)**. **Use StyleCop**

**3)**. **Use Nunit**

**4)**. Use Ncover

**Solution** :  
option [2] is correct

**Attempted** :  
option [3] is attempted

##### Q.**What will be the output of below code snippet?                                                                                                                                                          namespace NUnit.Tests {   using System;   using NUnit.Framework;   [TestFixture]   public class SuccessTests   {     [TestFixtureSetUp] public void Init()     { /\* ... \*/ }     [TestFixtureSetUp] public void Dispose()     { /\* ... \*/ }        } }**

**1)**. **It will run successfully**

**2)**. It will generate compilation error

**3)**. **It will run successfully but its tests will not run.**

**4)**. It will generate run time error

**Solution** :  
option [3] is correct

**Attempted** :  
option [1] is attempted

##### Q.**Ncover Code Coverage Report can be \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_formats**

**1)**. PDF

**2)**. EXCEL

**3)**. **HTML**

**4)**. **XML**

**Solution** :  
option [3,4] are correct

**Attempted** :  
option [4] is attempted

##### Q.**Seema requires NUnit implementation for her Online Shopping Mart application.She wants to  instruct the unit test engine that \_\_\_\_\_\_\_\_\_\_ method should be called after invoking each unit test.Which option would be appropriate?**

**1)**. **TearDown**

**2)**. SetUp

**3)**. Ignore

**4)**. **TextFixtureSetup**

**Solution** :  
option [1] is correct

**Attempted** :  
option [4] is attempted

#### Section 3: Solid Principles

##### Q.**A \_\_\_\_\_\_\_\_ is a special member function whose task is to initialize the objects of its class.**

**1)**. **Constructor**

**2)**. Destructor

**3)**. Selector

**4)**. Iterator

**Solution** :  
option [1] is correct

**Attempted** :  
option [1] is attempted

##### Q.**Identify which of the SOLID principle should be used to resolve the Issue in the code given below. public class ReportGeneration {     public string ReportType { get; set; }     public void GenerateReport(Employee em)     {         if (ReportType == "CRS")         {              // Report generation with employee data in Crystal Report.         }         if (ReportType == "PDF")         {             // Report generation with employee data in PDF.         }      } }**

**1)**. SRP - single responsibility principle

**2)**. **OCP - Open closed principle**

**3)**. LSP - Liskov substitution principle

**4)**. **ISP - Interface segregation principle**

**5)**. DIP - Dependency injection principle

**Solution** :  
option [2] is correct

**Attempted** :  
option [4] is attempted

##### Q.**A \_\_\_\_\_\_\_\_ is a description of a set of objects that share the same attributes, operations,  relationships, and semantics.**

**1)**. Structure

**2)**. **Class**

**3)**. Constructor

**4)**. **Function**

**Solution** :  
option [2] is correct

**Attempted** :  
option [4] is attempted

##### Q.**The process of compartmentalizing the elements of an abstraction that constitute its structure  and behavior is called as \_\_\_\_\_\_\_\_.**

**1)**. Hierarchy

**2)**. **Encapsulation**

**3)**. Modularity

**4)**. **Entity Abstraction**

**Solution** :  
option [2] is correct

**Attempted** :  
option [4] is attempted

##### Q.**Match the following: 1. SRP 2. LIS 3. DIP 4. OCP A. Parent should easily replace the child object B. The class is open for extensions, but closed for modifications  C. A class should have only one responsibility and not multiple D. The class delegates the dependent object creation to the client**

**1)**. 1-A, 2-D, 3-B, 4-C

**2)**. 1-B, 2-C, 3-A, 4-A

**3)**. **1-C, 2-A, 3-D, 4-B**

**4)**. 1-D, 2-B, 3-C, 4-D

**Solution** :  
option [3] is correct

**Attempted** :  
option [3] is attempted

##### Q.**Single inheritance, Multiple inheritance, and Aggregation comes under \_\_\_\_\_\_\_.**

**1)**. Modularity

**2)**. Typing

**3)**. **Hierarchy**

**4)**. Abstraction

**Solution** :  
option [3] is correct

**Attempted** :  
option [3] is attempted

##### Q.**Identify which of the SOLID principle is being violated in the code given below. public class Product {  public int Id{ get; set; }  public string Name{ get; set; }  public int Stock { get; set; }  public double Price { get; set; }   public void AddProduct(Product prodObj)  {   // Implementation    }  public void GenerateReport(Product prodObj)  {   // Implementation  } }**

**1)**. **SRP - single responsibility principle**

**2)**. OCP - Open closed principle

**3)**. LSP - Liskov substitution principle

**4)**. ISP - Interface segregation principle

**5)**. **DIP - Dependency injection principle**

**Solution** :  
option [1] is correct

**Attempted** :  
option [5] is attempted

##### Q.**Inheritance is an example of which type of hierarchy.**

**1)**. **Class Structure**

**2)**. Object Structure

**3)**. **Both**

**Solution** :  
option [1] is correct

**Attempted** :  
option [3] is attempted

##### Q.**The essential characteristics of an object that distinguish it from all other kinds of objects  and provide a conceptual boundary relative to viewers' perspective is called \_\_\_\_\_\_\_\_.**

**1)**. Encapsulation

**2)**. **Modularity**

**3)**. Hierarchy

**4)**. **Abstraction**

**Solution** :  
option [4] is correct

**Attempted** :  
option [2] is attempted

##### Q.**In which of the following mechanisms, types of all variables and expressions are fixed  at compilation time.**

**1)**. Strong Typing

**2)**. Weak Typing

**3)**. **Static Binding/ early binding**

**4)**. **Dynamic Binding/ late binding**

**Solution** :  
option [3] is correct

**Attempted** :  
option [4] is attempted

#### Section 4: C Sharp

##### Q.**What will be the output of the following code? static void Main(string[] args)         {             for (int i = 10; i > 0; i--)             {                 Console.Write(i + " ");                 i -= 1;             }             Console.ReadKey();         }**

**1)**. **10 9 8 7 6 5 4 3 2 1**

**2)**. 10 10 9 9 8 8 7 7 6 6 5 5 4 4 3 3 2 2 1 1

**3)**. **10 8 6 4 2**

**4)**. Infinite times 10

**Solution** :  
option [3] is correct

**Attempted** :  
option [1] is attempted

##### Q.**Which of the following is the .NET compatible event handlers form?**

**1)**. void Handler(EventArgs arg)  
{  
}

**2)**. **void Handler(object source, EventArgs arg)  
{  
}**

**3)**. void Handler(object source)  
{  
}

**4)**. void Handler(EventArgs arg, object source)  
{  
}

**Solution** :  
option [2] is correct

**Attempted** :  
option [2] is attempted

##### Q.**Which of the following type will access data using key?**

**1)**. ArrayList

**2)**. **Dictionaries**

**3)**. Vector Struct

**4)**. **Array**

**Solution** :  
option [2] is correct

**Attempted** :  
option [4] is attempted

##### Q.**Which of the following is used to release the raw memory back to the heap?**

**1)**. object finalization

**2)**. object deletion

**3)**. **object deallocation**

**4)**. object referencing

**Solution** :  
option [3] is correct

**Attempted** :  
option [3] is attempted

##### Q.**Indexers permit instances of a \_\_\_\_\_\_\_ to be indexed in the same way as arrays**

**1)**. **class only**

**2)**. structure only

**3)**. interface only

**4)**. **class or structure**

**Solution** :  
option [4] is correct

**Attempted** :  
option [1] is attempted

##### Q.**Which of the following statement will be true about below code snippet? int num1 = 300000; int num2 = 700000; long total = num1 + num2; Console.WriteLine(total);**

**1)**. Compile Time Error

**2)**. Exception will be thrown

**3)**. No error, but output will not be printed

**4)**. **Print : 1000000**

**Solution** :  
option [4] is correct

**Attempted** :  
option [4] is attempted

##### Q.**Predict the output of the following code:   class Test     {        static void Main(string[] args)         {             List<int> numbers = new List<int>();             numbers.Add(2);             numbers.Add(4);             numbers.Add(3);             foreach (int num in numbers)             {                 numbers.Remove(4);                 break;             }             foreach (int num in numbers)             {                 Console.WriteLine(num);             }         }     }**

**1)**. exception: collection modified in for each loop.

**2)**. prints:  
2  
4  
3

**3)**. **prints:  
2  
3**

**4)**. compilation error.

**Solution** :  
option [3] is correct

**Attempted** :  
option [3] is attempted

##### Q.**public void Multiply(double n1, double n2) {  Console.WriteLine((n1 \* n2).ToString()); } public void Addition(double n1, double n2) {  Console.WriteLine((n1 + n2).ToString()); }       Which of the following delegate will call both the methods given above?**

**1)**. public delegate MultiplyAddCalculatorDelegate;

**2)**. **public delegate double CalculatorDelegate(double n1, double n2);**

**3)**. **public delegate void CalculatorDelegate(paramarray double[]);**

**4)**. public delegate double CalculatorDelegate(paramarray double[]);

**Solution** :  
option [3] is correct

**Attempted** :  
option [2] is attempted

##### Q.**\_\_\_\_\_\_\_ attribute is used to mark the member data that should not participate in the serialization.**

**1)**. NonSerializable

**2)**. NonMember

**3)**. NonData

**4)**. **NonSerialized**

**Solution** :  
option [4] is correct

**Attempted** :  
option [4] is attempted

##### Q.**What will be the output of the following code snippet? class MyClass  {   char[] chrs = { 'A', 'B', 'C', 'D' };      public System.Collections.IEnumerator GetEnumerator()      {   foreach (char ch in chrs)                yield return ch;      }  }  class Program  {   static void Main(string[] args)      {   MyClass mc = new MyClass();          foreach (char ch in mc)               Console.Write(ch + " ");      }  }**

**1)**. Error in Code

**2)**. **Prints : A B C D**

**3)**. Exception

**4)**. Code will execute successfully, but will not print anything

**Solution** :  
option [2] is correct

**Attempted** :  
option [2] is attempted

Q.**What Will be the output of the following code?  
class Employee  
    {  
        public Employee()  
        {  
            Console.WriteLine("Employee Class Default Constructor");  
        }  
  
        public Employee(int EID)  
        {  
            Console.WriteLine("Employee Class Parameterized Constructor");  
        }  
    }  
  
    class Manager : Employee  
    {  
        public Manager()  
            : base()  
        {  
            Console.WriteLine("Manager Class Default Constructor");  
        }  
        public Manager(int EMPID, float da)  
            :base(EMPID)  
        {  
            Console.WriteLine("Manager Class Parameterized Constructor");  
        }  
    }  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            Manager mgr = new Manager();  
            Manager mgr1 = new Manager(2, 0.8f);  
            Console.ReadKey();  
        }  
    }**

**1)**. Employee Class Default Constructor  
Manager Class Default Constructor  
Employee Class Parameterized Constructor  
2  
0.8  
Manager Class Parameterized Constructor

**2)**. Manager Class Default Constructor  
Manager Class Parameterized Constructor

**3)**. **Employee Class Default Constructor  
Manager Class Default Constructor  
Employee Class Parameterized Constructor  
Manager Class Parameterized Constructor**

**4)**. Manager Class Default Constructor  
Employee Class Default Constructor  
Manager Class Parameterized Constructor  
Employee Class Parameterized Constructor

**Solution** :  
option [3] is correct

**Attempted** :  
option [3] is attempted

##### Q.**Check the following :  1. An anonymous method is, a block of code that is passed to a delegate. 2. An anonymous method consists of the keyword delegate, an optional parameter list, and a statement list enclosed in { and } delimiters 3. Anonymous methods can access ref or out parameters for defining methods 4. Anonymous methods are not working with events**

**1)**. **1 - True, 2 - True, 3- True, 4 - False**

**2)**. 1 - True, 2 - True, 3- False, 4 - True

**3)**. **1 - True, 2 - False, 3- False, 4 - False**

**4)**. 1 - True, 2 - True, 3- False, 4 - False

**Solution** :  
option [3] is correct

**Attempted** :  
option [1] is attempted

##### Q.**If user wish to use an Exception class in multiple catch statements, it should be the \_\_\_\_\_ catch statement.**

**1)**. first

**2)**. **last**

**3)**. second

**4)**. middle

**Solution** :  
option [2] is correct

**Attempted** :  
option [2] is attempted

##### Q.**If user needs to create an override of Object.Finalize then user needs to write \_\_\_\_\_**

**1)**. **Destructor**

**2)**. Delete

**3)**. Dispose

**4)**. Finalizer

**Solution** :  
option [1] is correct

**Attempted** :  
option [1] is attempted

##### Q.**The "using" keyword explicitly calls \_\_\_\_\_\_ methods at the end of the code block.**

**1)**. Close

**2)**. **Dispose**

**3)**. Destruct

**4)**. Discard

**Solution** :  
option [2] is correct

**Attempted** :  
option [2] is attempted

##### Q.**What will be the output of the following code? static void Calculate(ref int[] arr)         {             int i = 0;             foreach(var ele in arr)             {                 arr[i] = ele \* 5;                 i++;             }         }         static void Main(string[] args)         {             int[] arr = { 1, 2, 3, 4, 5 };             Calculate(ref arr);             foreach (var ele in arr)             {                 Console.Write(ele + " ");             }             Console.ReadKey();         }**

**1)**. 1 2 3 4 5

**2)**. 0 0 0 0 0

**3)**. 0 5 10 15 20

**4)**. **5 10 15 20 25**

**Solution** :  
option [4] is correct

**Attempted** :  
option [4] is attempted

##### Q.**Which of the following statements are true? Statement 1: Extension methods can be simple or static. Statement 2: Extension methods can help by enabling you to change a class without requiring the source code for the class.**

**1)**. **Statement 1 is true but Statement 2 is not true**

**2)**. **Statement 2 is true but Statement 1 is not true**

**3)**. Both the statements are true

**4)**. Both the statements are false

**Solution** :  
option [2] is correct

**Attempted** :  
option [1] is attempted

##### Q.**Which of the following codes will not compile?**

**1)**.  public struct Mystruct  
    {  
       public Mystruct(int a)  
        {            Console.WriteLine("This is mystruct constructor"+a);  
        }  
    }  
  
    class MainClass  
    {  
        static void Main(string[] args)  
        {  
            Mystruct s = new Mystruct(10);  
            Console.ReadLine();  
        }  
    }

**2)**. **public struct Mystruct  
    {  
       public Mystruct()  
        {            Console.WriteLine("This is mystruct constructor");  
        }  
    }  
  
    class MainClass  
    {  
        static void Main(string[] args)  
        {  
            Mystruct s = new Mystruct();  
            Console.ReadLine();  
        }  
    }**

**3)**.  public struct Mystruct  
    {  
       public Mystruct(int a)  
        {            Console.WriteLine("This is mystruct constructor"+a);  
        }  
    }  
  
    class MainClass  
    {  
        static void Main(string[] args)  
        {  
            Mystruct s = new Mystruct();  
            Mystruct s1 = new Mystruct(10);  
            Console.ReadLine();  
        }  
    }

**4)**. **public struct Mystruct  
    {  
       public Mystruct(int a)  
        {            Console.WriteLine("This is mystruct constructor");  
        }  
    }  
  
    class MainClass  
    {  
        static void Main(string[] args)  
        {  
            Mystruct s ;             
            Console.ReadLine();  
        }  
    }**

**Solution** :  
option [2] is correct

**Attempted** :  
option [4] is attempted

##### Q.**Match the following terms :**

### **Column A**

1. Interface

2. Abstract Class

### **Column B(Attempted)**

1. Contains the methods which are implemented by unrelated classes

2. Contains the methods which are implemented by a group of related classes

### **Solution**

1. Contains the methods which are implemented by unrelated classes

2. Contains the methods which are implemented by a group of related classes

##### Q.**Which of the following statements are true? Statement 1: Multicast delegates must contain only methods that return void. Statement 2: If the multicast delegate return any value then it is compile time error**

**1)**. **Statement 1 is true but Statement 2 is not true**

**2)**. Statement 2 is true but Statement 1 is not true

**3)**. Both the statements are true

**4)**. **Both the statements are false**

**Solution** :  
option [1] is correct

**Attempted** :  
option [4] is attempted

Q.**What will be the output of following code?  
  
namespace MulticastDel  
{  
    delegate void MathDelegate(int num1, int num2);  
      
    class MultcastDelegate  
    {  
        static void Add(int num1, int num2)  
        { Console.WriteLine("Calling Add()"); }  
        static void Subtract(int num1, int num2)  
        { Console.WriteLine("Calling Subtract()"); }  
  
        public static void Main()  
        {  
            Delegate\_Multicast func = new Delegate\_Multicast(Add);  
            func += new Delegate\_Multicast(Subtract);  
            func(1,2);               
            func -= new Delegate\_Multicast(Add);  
            func(2,3);               
            Console.ReadLine();  
        }     
    }  
}**

**1)**. Calling Add()  
Calling Add()  
Calling Subtract()

**2)**. **Calling Add()  
Calling Subtract()  
Calling Subtract()**

**3)**. **Calling Subtract()  
Calling Subtract()**

**4)**. Calling Add()  
Calling Subtract()

**Solution** :  
option [2] is correct

**Attempted** :  
option [3] is attempted

##### Q.**What will be the output of the below code snippet? static void Main(string[] args) {    int num = 5;    for(int i = 1; i <= 5; i++)            num = num \* i;    Console.WriteLine(num); }**

**1)**. **600**

**2)**. 25

**3)**. 120

**4)**. **5**

**Solution** :  
option [1] is correct

**Attempted** :  
option [4] is attempted

##### Q.**Which of the following statements are correct about Shared assemblies?**

**1)**. **These are intended to be common libraries that specific application can use**

**2)**. **Shared assemblies are stored in Global Assembly Cache**

**3)**. These assemblies do not require to maintain version information

**4)**. **A shared assembly must have strong name**

**Solution** :  
option [2,4] are correct

**Attempted** :  
option [1,2,4] are attempted

##### Q.**Following is the feature of \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_. 1. Associate declarative information with C# code components 2. Based on this, runtime can change its behavior or course of action**

**1)**. **Attributes**

**2)**. **Reflection**

**3)**. Assembly

**4)**. Generics

**Solution** :  
option [1] is correct

**Attempted** :  
option [2] is attempted

##### Q.**\_\_\_\_\_\_\_\_contains all the metadata with the  1. Version of assemblies 2. Scope of assemblies 3. Reference to resource and class 4. Stored in Portable Executable file (PE) or in MSIL.**

**1)**. Metadata

**2)**. **Mainfest**

**3)**. Assembly

**4)**. MSIL

**Solution** :  
option [2] is correct

**Attempted** :  
option [2] is attempted

Q.**Which of the following statements are true?  
Statement 1:  
Directory class exposes static methods for creating, moving, and enumerating through directories and subdirectories.  
Statement 2:  
DirectoryInfo class exposes instance methods for creating, moving, and enumerating through directories and subdirectories.**

**1)**. Statement 1 is true but Statement 2 is not true

**2)**. Statement 2 is true but Statement 1 is not true

**3)**. **Both the statements are true**

**4)**. Both the statements are false

**Solution** :  
option [3] is correct

**Attempted** :  
option [3] is attempted

##### Q.**The -------------- class is inherited to create a user defined attribute.**

**1)**. **System.CustomAttribute.**

**2)**. System.SystemAttribute.

**3)**. **System.Attribute.**

**4)**. System.Reflection.Type.

**Solution** :  
option [3] is correct

**Attempted** :  
option [1] is attempted

##### Q.**To enable the class to initialize a nonserialized member automatically, which of the following interfaces can be used?**

**1)**. **IDeserializationCallBack**

**2)**. OnDeserialization

**3)**. IDeserializationBack

**4)**. IserializationCallBack

**Solution** :  
option [1] is correct

**Attempted** :  
option [1] is attempted

Q.**Which of the following statements are correct?  
1. To provide stronger compile-time type checking and reduce type casts, C# permits an optional list of constraints to be supplied for each type parameter.  
2. Constraints can not be specified on generics.  
3. Constraints are declared using the word where, followed by the name of a type parameter, followed by a list of class or interface types.  
4. A type parameter constraint specifies a requirement that a type must fulfill in order to be used as an argument for that type parameter.**

**1)**. 1 - False, 2 - False, 3 - True, 4 - True

**2)**. 1 - True, 2 - False, 3 - False, 4 - True

**3)**. **1 - True, 2 - False, 3 - True, 4 - True**

**4)**. 1 - True, 2 - False, 3 - True, 4 - False

**Solution** :  
option [3] is correct

**Attempted** :  
option [3] is attempted

##### Q.**A Hashtable dept maintains the collection of department id and department name in the company. Which of the following is the correct way to find out whether "101" department id is present in the collection or not?**

**1)**. **dept.ContainsKey(101);**

**2)**. **dept.HasValue(101);**

**3)**. dept.HasKey(101);

**4)**. dept.ContainsValue(101);

**Solution** :  
option [1] is correct

**Attempted** :  
option [2] is attempted

##### Q.**What will be the output of the following code? static void Main(string[] args)         {             const int num1 = 2;             const int num2 = 3;             for (var i = 0; i < 5; i++)             {                 num1 = num1 \* i;                 num2 = num2 \* i;             }             Console.WriteLine(num1 + " " + num2);             Console.ReadKey();         }**

**1)**. 0 0

**2)**. **2 3**

**3)**. **Compile Time Error**

**4)**. 8 12

**Solution** :  
option [3] is correct

**Attempted** :  
option [2] is attempted

##### Q.**Which of the following class provides the base class for both FileInfo and DirectoryInfo objects?**

**1)**. File

**2)**. **Directory**

**3)**. **FileSystemInfo**

**4)**. Path

**Solution** :  
option [3] is correct

**Attempted** :  
option [2] is attempted

##### Q.**Which of the following are valid enumeration values of File Access?**

**1)**. **Update**

**2)**. **Read**

**3)**. **ReadWrite**

**4)**. **Delete**

**5)**. **Write**

**Solution** :  
option [2,3,5] are correct

**Attempted** :  
option [1,2,4,5] are attempted

##### Q.**Which of the following utility will be used yo provide a strong name for an assembly  and generate the public / private key data?**

**1)**. **sn.exe**

**2)**. **gacutil.exe**

**3)**. ildasm.exe

**4)**. None of these

**Solution** :  
option [1] is correct

**Attempted** :  
option [2] is attempted

Q.**What will be the output of the given code snippet?  
  
static int Sum(int num1, int num2)  
        {  
            int ans;  
  
            num1 = 100;  
            num2 = 80;  
            ans = num1 + num2;  
              
            return ans;  
        }  
  
        static void Main(string[] args)  
        {  
            int num1, num2;  
  
            num1 = 5;  
            num2 = 3;  
  
            Console.WriteLine(num1 + " " + num2);  
            Console.WriteLine(Sum(num1, num2));  
            Console.WriteLine(num1 + " " + num2);  
Console.ReadKey();  
        }**

**1)**. 5 3  
8  
100 80

**2)**. 5 3  
8  
5 3

**3)**. 5 3  
180  
100 80

**4)**. **5 3  
180  
5 3**

**Solution** :  
option [4] is correct

**Attempted** :  
option [4] is attempted

##### Q.**What will be the output of the below code snippet? static void Main(string[] args) {    int i, j;    int[ , ] arr = new int[3, 3];    for(i = 0; i < 3; i++)    {         for(j = 0; j < 3; j++)         {              arr[i, j] = i \* 2 + j \* 2;              Console.WriteLine(arr[i, j] + " ");         }         Console.WriteLine();    } }**

**1)**. **0 2 4  
2 4 6  
4 6 8**

**2)**. **0 0 0  
2 2 2  
4 4 4**

**3)**. 0 4 16  
4 16 32  
16 32 64

**4)**. 0 0 0  
4 4 4  
8 8 8

**Solution** :  
option [1] is correct

**Attempted** :  
option [2] is attempted

##### Q.**Which of the following is the correct way to declare events?**

**1)**. public event void MyClickEvent;

**2)**. delegate void MyHandler();  
public class EventClass  
{  
      public event void MyClickEvent;  
}

**3)**. **delegate void MyHandler();  
public class EventClass  
{  
      public event MyHandler MyClickEvent;  
}**

**4)**. **delegate void MyHandler();  
public class EventClass  
{  
      public MyHandler event MyClickEvent;  
}**

**Solution** :  
option [3] is correct

**Attempted** :  
option [4] is attempted

##### Q.**Which of the following collection class will allow the user to access the element based on key?**

**1)**. SortedList

**2)**. **HashTable**

**3)**. ArrayList

**4)**. Stack

**Solution** :  
option [2] is correct

**Attempted** :  
option [2] is attempted

Q.**What will be the output of the following code?  
  
 class Program  
    {  
        void Addition(int num1, int num2)  
        {  
            Console.WriteLine("Calling Integer Method");  
        }  
        void Addition(long num1, long num2)  
        {  
            Console.WriteLine("Calling Long Integer Method");  
        }  
        void Addition(short num1, short num2)  
        {  
            Console.WriteLine("Calling Short Integer Method");  
        }  
       static void Main(string[] args)  
        {  
            new Test().Addition(3, 4);  
        }  
    }**

**1)**. **Calling Integer Method**

**2)**. Calling Short Integer Method

**3)**. Calling Long Integer Method

**4)**. Calling Integer Method  
Calling Long Integer Method  
Calling Short Integer Method

**Solution** :  
option [1] is correct

**Attempted** :  
option [1] is attempted

##### Q.**A try block must be followed by \_\_\_\_\_**

**1)**. **catch only**

**2)**. finally only

**3)**. **catch or finally**

**4)**. throw

**Solution** :  
option [3] is correct

**Attempted** :  
option [1] is attempted

##### Q.**Which of the following statement/s is/are correct? Statement 1 : Delegates are preferred method of implementing callbacks in the CLR Statement 2 : By referencing a method as a delegate, the method will be treated as object**

**1)**. Statement 1 is true but Statement 2 is false

**2)**. Statement 2 is true but Statement 1 is false

**3)**. **Both statements are true**

**4)**. Both statements are false

**Solution** :  
option [3] is correct

**Attempted** :  
option [3] is attempted

##### Q.**Vicky has written below code for anonymous method : public delegate string StringDelegate(string str); public static void Main() {        StringDelegate del = delegate(string str)        {                return str.ToUpper();        };        Console.WriteLine(del(".Net batch"); } Which of the following code snippet will give same output as that of above code?**

**1)**. StringDelegate del => { return str.ToUpper(); };  
Console.WriteLine(del);

**2)**. **StringDelegate del = str => { return str.ToUpper(); };  
Console.WriteLine(del);**

**3)**. StringDelegate del => str = { return str.ToUpper(); };  
Console.WriteLine(del);

**4)**. StringDelegate del = str = { return str.ToUpper(); };  
Console.WriteLine(del);

**Solution** :  
option [2] is correct

**Attempted** :  
option [2] is attempted

##### Q.**Which of the following statement is not correct about IEnumerable interface?**

**1)**. Each collection class must implement IEnumerable interface

**2)**. **IEnumerable is not inherited by ICollection**

**3)**. The IEnumerable interface has one method called the GetEnumerator() method

**4)**. Class implementing GetEnumerator() method must return a class that implements the IEnumerator interface

**Solution** :  
option [2] is correct

**Attempted** :  
option [2] is attempted

##### Q.**Which  interface must be implemented when a class wants to sort its instances?**

**1)**. ICompare

**2)**. **IComparable**

**3)**. IList

**4)**. IDisposable

**Solution** :  
option [2] is correct

**Attempted** :  
option [2] is attempted

##### Q.**Which of the following statement/s is/are correct? Statement 1 : A block of code on the caller's side that invokes the exception-prone member is known as catch block Statement 2 : A block of code on the caller's side that processes the exception is known as try block**

**1)**. **Statement 1 is true but Statement 2 is false**

**2)**. Statement 2 is true but Statement 1 is false

**3)**. Both statements are true

**4)**. **Both statements are false**

**Solution** :  
option [4] is correct

**Attempted** :  
option [1] is attempted

##### Q.**Check the following :  1. Parallel extension set of libraries and tools makes concurrent programming easier. 2. Parallel extension set of libraries and tools makes profiling and debugging concurrent applications easier. 3. Parallel extension set of libraries and tools enables the applications to scale up without any code as number of cores increases 4. Parallel extension set of libraries and tools do not allow existing applications to be easily parallelized**

**1)**. **1 - True, 2 - True, 3 -True, 4 - False**

**2)**. 1 - True, 2 - False, 3 -True, 4 - False

**3)**. **1 - True, 2 - True, 3 -False, 4 - False**

**4)**. 1 - True, 2 - False, 3 -False, 4 - False

**Solution** :  
option [1] is correct

**Attempted** :  
option [3] is attempted

##### Q.**Which of the following statements are correct about try block?**

**1)**. **Try block can be followed by 0 or more catch blocks**

**2)**. **Try block can be followed by finally instead of catch**

**3)**. **Try block can be followed by both catch and finally**

**4)**. Try block need not be followed by anything

**Solution** :  
option [1,2,3] are correct

**Attempted** :  
option [2,3] are attempted

##### Q.**State whether the following statements are true w.r.t the Enumerator object:   a. Enumerator provides read-only, forward-only cursor for set of items. b. If the collection to which the enumerator is referencing is modified, the enumeration      may not work. c. The GetEnumerator() method returns an instance of the IDictionary class d. The Reset property of the enumerator sets the enumerator position to the current      element of the collection. e. Classes that implement the GetEnumerator() must return a class that  implements       the IEnumerator interface.**

**1)**. **A-true,B-true,C-false,D-false,E-true**

**2)**. A-true,B-true,C-true,D-true,E-true

**3)**. A-false,B-false,C-false,D-false,E-false

**4)**. A-true,B-true,C-false,D-false,E-false

**Solution** :  
option [1] is correct

**Attempted** :  
option [1] is attempted

##### Q.**Which of the following statements are false?**

**1)**. When a publisher class raises an event, all the subscribing classes are notified

**2)**. **When a subscribing class raises an event, all the publishing classes are notified**

**3)**. **How an event is handled, is the responsibility of the subscriber class**

**4)**. **delegates are the functions used to handle the events raised in a program**

**Solution** :  
option [2,4] are correct

**Attempted** :  
option [2,3,4] are attempted

##### Q.**Which of the following is the correct code for Generic Delegate?**

**1)**. **delegate void StackEventHandler<T, U>(T sender, U eventArgs);  
  
class Stack<T>  
{  
public class StackEventArgs : System.EventArgs { }  
public event StackEventHandler<Stack<T>, StackEventArgs> stackEvent;  
  
protected virtual void OnStackChanged(StackEventArgs a)  
{  
stackEvent(this, a);  
}  
}  
  
class SampleClass  
{  
public void HandleStackChange<T>(Stack<T> stack, Stack<T>.StackEventArgs args) { }  
}  
  
public static void Test()  
{  
Stack<double> s = new Stack<double>();  
SampleClass o = new SampleClass();  
s.stackEvent += o.HandleStackChange;  
}**

**2)**. delegate void StackEventHandler(T sender, U eventArgs);  
  
class Stack<T>  
{  
public class StackEventArgs : System.EventArgs { }  
      
protected virtual void OnStackChanged(StackEventArgs a)  
{  
stackEvent(this, a);  
}  
}  
  
class SampleClass  
{  
public void HandleStackChange<T>(Stack<T> stack, Stack<T>.StackEventArgs args) { }  
}  
  
public static void Test()  
{  
Stack<double> s = new Stack<double>();  
SampleClass o = new SampleClass();  
s.stackEvent += o.HandleStackChange;  
}  
  
+H36

**3)**. delegate void StackEventHandler<T, U>(T sender, U eventArgs);  
  
class Stack<T>  
{  
public class StackEventArgs : System.EventArgs { }  
public event StackEventHandler<Stack<T>, StackEventArgs> stackEvent;  
  
protected virtual void OnStackChanged(StackEventArgs a)  
{  
stackEvent(this, a);  
}  
}  
  
class SampleClass  
{  
public void HandleStackChange<T>(Stack<T> stack, Stack<T>.StackEventArgs args) { }  
}  
  
public static void Test()  
{  
Stack<double> s = new Stack<double>();  
SampleClass o = new SampleClass();  
s.stackEvent -= o.HandleStackChange;  
}

**4)**. delegate void StackEventHandler<T, U>(T sender, U eventArgs);  
  
class Stack<T>  
{  
public class StackEventArgs : System.EventArgs { }  
public event StackEventHandler<Stack<T>, StackEventArgs> stackEvent;  
  
protected virtual void OnStackChanged(StackEventArgs a)  
{  
stackEvent(this, a);  
}  
}  
  
class SampleClass  
{  
public void HandleStackChange<T>(Stack<T> stack, Stack<T>.StackEventArgs args) { }  
}  
  
public static void Test()  
{  
Stack<double> s = new Stack<double>();  
SampleClass o = new SampleClass();  
s.stackEvent += o.HandleStackChange<T>;  
}

**Solution** :  
option [1] is correct

**Attempted** :  
option [1] is attempted

##### Q.**Predict the output of the following code:   class Test     {         void TestMethod()         {             throw new Exception();         }        static void Main(string[] args)         {             try             {                 new Test().TestMethod();             }             catch (SystemException)             {                 Console.WriteLine("some exception");             }                     }     }**

**1)**. **program crashes due to unhandled exception.**

**2)**. prints "some exception".

**3)**. compilation error.

**4)**. executes successfully.

**Solution** :  
option [1] is correct

**Attempted** :  
option [1] is attempted

##### Q.**Which of the following statement is not correct about attributes?**

**1)**. Attribute is a sub class of System.Attribute

**2)**. Attributes can be applied to various code parts in .net and are called Targets for an attribute

**3)**. Attributes concept in .Net is a way to mark or store meta data about the code in assembly

**4)**. **The Runtime can not change its behavior or course of action based on the attribute present**

**Solution** :  
option [4] is correct

**Attempted** :  
option [4] is attempted

##### Q.**Which of the following are the correct ways to declare the nullable integer variable?**

**1)**. **Nullable int**

**2)**. **Nullable<Int32>**

**3)**. Nullable Int32

**4)**. int Nullable

**5)**. **int? Nullable**

**Solution** :  
option [2,5] are correct

**Attempted** :  
option [1,2,5] are attempted

##### Q.**In the Fxcop Analysis result, what does Absent signify?**

**1)**. **Issues that have been resolved**

**2)**. **Issues that have been disabled**

**3)**. Issues are not resolved

**4)**. Issues does not exists

**Solution** :  
option [1] is correct

**Attempted** :  
option [2] is attempted

##### Q.**Match the following parameter types:**

### **Column A**

1. out

2. ref

### **Column B(Attempted)**

1. It should be assigned before passing to method

2. It can be unassigned initially within a method

### **Solution**

1. It can be unassigned initially within a method

2. It should be assigned before passing to method

##### Q.**Consider the code given below.     class Program     {         static void LowerCase(string str)         {             Console.WriteLine(str.ToLower());         }         static void UnaryAddition(int num)         {             Console.WriteLine(1 + num++ + 1);         } } Which of the following delegate will be used to refer/call both the methods given in the code?**

**1)**. delegate void MyGenericDelegate(T arg);

**2)**. **delegate void MyGenericDelegate<T>(T arg);**

**3)**. **delegate void MyGenericDelegate<int, string>(T arg);**

**4)**. delegate void MyGenericDelegate<T>(arg);

**Solution** :  
option [2] is correct

**Attempted** :  
option [3] is attempted

##### Q.**Which of the following statements are correct about finally block?**

**1)**. **Finally block encapsulates the code that cleans up any resources**

**2)**. **Finally block is executed only if exception is thrown**

**3)**. **Finally block is optional**

**4)**. A code can contain multiple finally block

**Solution** :  
option [1,3] are correct

**Attempted** :  
option [1,2,3] are attempted

##### Q.**Which of the following statements are true about Static Methods?**

**1)**. **Static methods have static keyword as the method definition**

**2)**. **Static methods are accessed with creating an instance of the class object**

**3)**. **Static methods will not have "this" functionality to access the method**

**4)**. **Static methods cannot access non static members of the class.**

**Solution** :  
option [1,3,4] are correct

**Attempted** :  
option [1,2,4] are attempted

##### Q.**Which of the following statement is correct about the below code snippet? Var num;**

**1)**. **Compile Time Error : Variable must be initialized**

**2)**. Runtime Error;

**3)**. Null value is assigned to num

**4)**. 0 is assigned to num

**Solution** :  
option [1] is correct

**Attempted** :  
option [1] is attempted

##### Q.**Which of the following statements are true? Statement 1: Output parameters must be initialized before they are passed to the method. Statement 2: Reference parameters do not need to be initialized before they are passed to the method.**

**1)**. Statement 1 is true but Statement 2 is not true

**2)**. Statement 2 is true but Statement 1 is not true

**3)**. **Both the statements are true**

**4)**. **Both the statements are false**

**Solution** :  
option [4] is correct

**Attempted** :  
option [3] is attempted

##### Q.**Which of the following is the correct way to load the assembly and obtain a list of various types in the assembly?**

**1)**. **Assembly maths = Assembly.LoadFrom("Maths.dll");  
Types[] mathType = maths.GetTypes();**

**2)**. Assembly maths = Assembly.LoadFrom("Maths.dll");  
Types[] mathType = maths.GetType();

**3)**. **Assembly maths = Assembly.Load("Maths.dll");  
Types[] mathType = maths.GetTypes();**

**4)**. Assembly maths = Assembly.Load("Maths.dll");  
Types[] mathType = maths.GetType();

**Solution** :  
option [1] is correct

**Attempted** :  
option [3] is attempted

##### Q.**What will be the output of the following code? static void Main(string[] args)         {             string str1 = "CAPGEMINI CORPORATE UNIVERSITY";             string substr;             substr = str1.Substring(10, 4);             Console.WriteLine(substr);             Console.ReadKey();         }**

**1)**. **CORP**

**2)**. CORPORATE

**3)**.  CORPORATE

**4)**. **CORP**

**Solution** :  
option [4] is correct

**Attempted** :  
option [1] is attempted

##### Q.**Which of the following statement is/are not correct about yield?**

**1)**. **When the yield return statement is reached, the previous location is stored.**

**2)**. **The yield keyword is used to specify the value, or values, returned.**

**3)**. **It is used in an iterator block to provide a value to the enumerator object or to signal the end of iteration**

**4)**. **The return statement is as below:  
return yield <expression>;**

**Solution** :  
option [1,4] are correct

**Attempted** :  
option [2,3] are attempted

##### Q.**What will be the output of the below code snippet? static void Main(string[] args) {    string str = "SPEAK";    string revStr = "";    int length = str.Length - 1;    while(length >= 0)    {        revStr = revStr + str[length];        length--;    }    Console.WriteLine(revStr); }**

**1)**. **SPEA**

**2)**. **KAEPS**

**3)**. AEPS

**4)**. KAEP

**Solution** :  
option [2] is correct

**Attempted** :  
option [1] is attempted

##### Q.**Which of the following is the correct representation for extension method on String class?**

**1)**. public class StringExtension  
{  
public static string SplitVowels(this string str)  
{ ... }  
}

**2)**. **public static class StringExtension  
{  
public static string SplitVowels(this string str)  
{ ... }  
}**

**3)**. public class StringExtension  
{  
public static string SplitVowels(string str)  
{ ... }  
}

**4)**. **public static class StringExtension  
{  
public static string SplitVowels(string str)  
{ ... }  
}**

**Solution** :  
option [2] is correct

**Attempted** :  
option [4] is attempted

Q.**What will be the output of the following code?  
  
class Employee  
    {  
        int empid;  
        string name;  
  
        public int EmpId { get { return empid; } set { empid = value; } }  
        public string Name { get { return name; } set { name = value; } }  
  
        public Employee()  
        {  
            empid = 0;  
            name = "null";  
        }  
  
        public Employee(int empid, string name)  
        {  
            empid = empid;  
            name = name;  
        }  
    }  
  
    class Manager : Employee  
    {  
        double salary;  
  
        public double Salary { get { return salary; } set { salary = value; } }  
  
        public Manager()  
        {  
            salary = 0.0;  
        }  
  
        public Manager(int empid, string name, double salary)  
        {  
            salary = salary;  
        }  
    }  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            Manager mgr = new Manager();  
  
            Console.WriteLine("Emp ID : {0} Name : {1} Salary : {2}", mgr.EmpId, mgr.Name, mgr.Salary);  
  
            Manager mgr1 = new Manager(2, "John", 30000);  
  
            Console.WriteLine("Emp ID : {0} Name : {1} Salary : {2}", mgr.EmpId, mgr.Name, mgr.Salary);  
  
            Console.ReadKey();  
        }  
    }**

**1)**. **Emp ID : 0 Name : null Salary : 0  
Emp ID : 0 Name : null Salary : 0**

**2)**. Emp ID : 0 Name : null Salary : 0.0  
Emp ID : 0 Name : null Salary : 0.0

**3)**. **Emp ID : 0 Name : null Salary : 0.0  
Emp ID : 0 Name : null Salary : 30000**

**4)**. Emp ID : 0 Name : null Salary : 0.0  
Emp ID : 2 Name : John Salary : 30000

**Solution** :  
option [1] is correct

**Attempted** :  
option [3] is attempted

##### Q.**What will be the output of below code snippet? static void Main(string[] args) {      int i = 5;      int j;      MyMethod1(ref i);      MyMethod2(out j);      Console.writeline(i + "  " + j); } static void MyMethod1(ref int x) {      x = x + x; } static void MyMethod2(out int x) {     x = 6;     x = x \* x; }**

**1)**. 36 10

**2)**. **10 36**

**3)**. 5 0

**4)**. 5 6

**Solution** :  
option [2] is correct

**Attempted** :  
option [2] is attempted

##### Q.**Which of the following keyword will be used in iterator to return the element or to break?**

**1)**. With

**2)**. **Yield**

**3)**. Optional

**4)**. Using

**Solution** :  
option [2] is correct

**Attempted** :  
option [2] is attempted

##### Q.**Which of the following statement is not correct about a C# class?**

**1)**. Class is a template that defines the form of an object

**2)**. Class is a set of plans that specify how to build an object

**3)**. **Physical representation of class exist in memory when you just declare the object**

**4)**. **C# is a reference type**

**Solution** :  
option [3] is correct

**Attempted** :  
option [4] is attempted

##### Q.**Predict the output of the following code:  public class Test     {         public static I Enumerable Get Data()         {             yield return 'Y';              yield return 'E'; //line 1             yield return 'S';         }                  static void Main(string[] args)         {             string str = "";             for each (char c in Get Data()) //line 2             {                 str = str + c;              }             Console.WriteLine(str);         }     }**

**1)**. **prints YES**

**2)**. prints Y

**3)**. prints E

**4)**. prints S

**5)**. **Compile time error at line 1**

**6)**. Compile time error at line 2

**Solution** :  
option [1] is correct

**Attempted** :  
option [5] is attempted

##### Q.**Predict the output of the following code. static void Main(string[] args)         {             Console.WriteLine("Char : {0}", sizeof(char));             Console.WriteLine("Boolean : {0}", sizeof(bool));             Console.WriteLine("Unsigned Short : {0}", sizeof(short));             Console.WriteLine("Single : {0}", sizeof(Single));             Console.ReadKey();         }**

**1)**. **Char : 2  
Boolean : 1  
Unsigned Short : 2  
Single : 4**

**2)**. **Char : 1  
Boolean : 1  
Unsigned Short : 2  
Single : 4**

**3)**. Char : 2  
Boolean : 1  
Unsigned Short : 2  
Single : 2

**4)**. Char : 1  
Boolean : 1  
Unsigned Short : 2  
Single : 2

**Solution** :  
option [1] is correct

**Attempted** :  
option [2] is attempted

##### Q.**Which of the following statements are correct about Exception?**

**1)**. Exceptions are typically regarded as compile time anomalies

**2)**. **Exception handling is a technique  
for dealing with runtime exceptions.**

**3)**. Exceptions are notifications that some syntax errors have occurred in program.

**4)**. **Exception is an event that occurs during the execution of a program that disrupts the normal flow of instructions during the execution of program.**

**Solution** :  
option [2,4] are correct

**Attempted** :  
option [2,4] are attempted

##### Q.**Which of the following statement/s is/are correct? Statement 1 : If the object overrides the Object.Finalize method then this method will be called by the garbage collector. Statement 2 : struct cannot override the Finalize method.**

**1)**. **Statement 1 is true but Statement 2 is false**

**2)**. Statement 2 is true but Statement 1 is false

**3)**. **Both statements are true**

**4)**. Both statements are false

**Solution** :  
option [3] is correct

**Attempted** :  
option [1] is attempted

##### Q.**Consider the below given code and identify the output generated. class Employee         {             static Employee()             {                 Console.WriteLine("Static Employee Constructor");             }             public static void Display()             {                 Console.WriteLine("Employee Display Method");             }         }         static void Main(string[] args)         {             Employee.Display();                          Console.ReadKey();         }**

**1)**. Employee Display Method

**2)**. **Static Employee Constructor  
Employee Display Method**

**3)**. Static Employee Constructor

**4)**. Employee Display Method  
Static Employee Constructor

**Solution** :  
option [2] is correct

**Attempted** :  
option [2] is attempted

##### Q.**Which of the following statements are true? Statement 1: If user don't provide any constructor, the compiler will generate a default constructor. Statement 2: The default constructor created by compiler is just initializes reference variables to null, value type variables to zero and boolean variables to false.**

**1)**. Statement 1 is true but Statement 2 is not true

**2)**. **Statement 2 is true but Statement 1 is not true**

**3)**. **Both the statements are true**

**4)**. Both the statements are false

**Solution** :  
option [3] is correct

**Attempted** :  
option [2] is attempted

##### Q.**Which of the following statements are true in case of an Interface?**

**1)**. **Each class that includes an interface must implement all of the methods**

**2)**. An Interface can contain only Implementation and no Abstract methods at all

**3)**. **An Interface can contain only Abstract methods and no Implementation at all**

**4)**. **An Interface can not be Inherited by other interface**

**Solution** :  
option [1,3] are correct

**Attempted** :  
option [1,3,4] are attempted

##### Q.**Which of the following property will give number of elements from array?**

**1)**. Count

**2)**. **Length**

**3)**. Depth

**4)**. NumberCount

**Solution** :  
option [2] is correct

**Attempted** :  
option [2] is attempted

##### Q.**Which of the following class listens to the file system change notifications and raises events  when a directory, or file in a directory, changes?**

**1)**. **FileSystemWatcher**

**2)**. File

**3)**. Directory

**4)**. **FileSystemInfo**

**Solution** :  
option [1] is correct

**Attempted** :  
option [4] is attempted

##### Q.**Which of the following method opens a specified file, writes out an array of strings, and then closes the file?**

**1)**. WriteAllBytes()

**2)**. **WriteAllLines()**

**3)**. WriteAllText()

**4)**. WriteAllStrings()

**Solution** :  
option [2] is correct

**Attempted** :  
option [2] is attempted

##### Q.**Which of the following statement are correct about TPL?**

**1)**. **Task parallelism refers to one or more independent tasks running concurrently**

**2)**. **TPL is based on the concept of the task**

**3)**. **A task represents an asynchronous operation**

**4)**. It is not efficient to use system resource

**Solution** :  
option [1,2,3] are correct

**Attempted** :  
option [1,2,3] are attempted

**Q.** Which of the following statement is correct about the below code snippet?  
  
Var marks = {"aa",10,2.3};

**1)**. The collection of values are assigned to marks

**2)**. Compile Time Error : Can not covert implicitly

**3)**. **Compile Time Error : Can not initialize variable with array initializer**

**4)**. **Runtime Error**

**Solution** :  
option [3] is correct

**Attempted** :  
option [4] is attempted

##### Q.**Which of the following is the correct way to assign annonymous method to delegate and invoke it?**

**1)**. **delegate string MyDelegate(string str);  
  
MyDelegate del = delegate(string str)  
{ return str.ToUpper(); }  
Console.WriteLine(del(".net"));**

**2)**. delegate string MyDelegate(string str);  
  
MyDelegate del = delegate(str)  
{ return str.ToUpper(); }  
Console.WriteLine(del(".net"));

**3)**. delegate string MyDelegate(string str);  
  
MyDelegate del = delegate(string str)  
return str.ToUpper();   
Console.WriteLine(del(".net"));

**4)**. delegate string MyDelegate(string str);  
  
MyDelegate del = new MyDelegate(string str)  
{ return str.ToUpper(); }  
Console.WriteLine(del(".net"));

**Solution** :  
option [1] is correct

**Attempted** :  
option [1] is attempted

##### Q.**Which of the following class provides properties and instance methods for the creation, copying,  deletion, moving and opening of files?**

**1)**. File

**2)**. **FileInfo**

**3)**. Directory

**4)**. DirectoryInfo

**Solution** :  
option [2] is correct

**Attempted** :  
option [2] is attempted

##### Q.**Which of the following is the correct property to find out the maximum limit of elements that can be present in an ArrayList?**

**1)**. GrowSize

**2)**. **Capacity**

**3)**. MaxCount

**4)**. Count

**Solution** :  
option [2] is correct

**Attempted** :  
option [2] is attempted

##### Q.**Which of the following statement/s is/are correct? Statement 1 : Lambda expressions passed as arguments participate in type argument inference and in method overload resolution. Statement 2 : Lambda expressions with an expression body can be converted to expression trees.**

**1)**. Statement 1 is true but Statement 2 is false

**2)**. Statement 2 is true but Statement 1 is false

**3)**. **Both statements are true**

**4)**. Both statements are false

**Solution** :  
option [3] is correct

**Attempted** :  
option [3] is attempted

##### Q.**Which of the following method will allow you to load an assembly using it's file name or path?**

**1)**. Assembly.GetExecutingAssembly()

**2)**. **Assembly.LoadFrom()**

**3)**. Assembly.GetAssembly()

**4)**. Assembly.LoadModule()

**Solution** :  
option [2] is correct

**Attempted** :  
option [2] is attempted

##### Q.**Which of the following statement/s is/are correct? Statement 1 : The value passed for a params parameter can be either a comma separated value list or a single dimensional array Statement 2 : In method there can be multiple params parameter**

**1)**. **Statement 1 is true but Statement 2 is false**

**2)**. Statement 2 is true but Statement 1 is false

**3)**. Both statements are true

**4)**. Both statements are false

**Solution** :  
option [1] is correct

**Attempted** :  
option [1] is attempted

##### Q.**If user determine that a given class has some member data that should not participate in the serialization scheme, then which of the following attribute is used for the member?**

**1)**. **[NonSerialization]**

**2)**. [NonSerializable]

**3)**. **[NonSerialized]**

**4)**. [NonSerialize]

**Solution** :  
option [3] is correct

**Attempted** :  
option [1] is attempted

##### Q.**What will be the output of below code snippet? class A  {   public int i;      public void Display()       {   Console.WriteLine(i);     }  }      class B: A   {   public int j;      public void Display()       {   Console.WriteLine(j);     }  }      class Program  {   static void Main(string[] args)      {          B obj = new B();          obj.i = 1;          obj.j = 2;          obj.Display();       }  }**

**1)**. 0

**2)**. **2**

**3)**. 1

**4)**. 1 2

**Solution** :  
option [2] is correct

**Attempted** :  
option [2] is attempted

##### Q.**Check the following :  1. Dispose method cannot call Finalize. 2. Finalize method can call Dispose. 3. User can call Finalize method. 4. User cannot call Dispose method. 5. Finalize is protected method.**

**1)**. 1 - True, 2 - True, 3 - True, 4 - False, 5 - True

**2)**. **1 - True, 2 - True, 3 - False, 4 - False, 5 - True**

**3)**. 1 - True, 2 - True, 3 - False, 4 - False, 5 - False

**4)**. 1 - True, 2 - True, 3 - False, 4 - True, 5 - True

**Solution** :  
option [2] is correct

**Attempted** :  
option [2] is attempted

##### Q.**Read the following code carefully: using System; public class Test {    public int mark=0;     public int Marks       {   get { return mark; }      } } public class Program {       public static void Main()       {         Test t = new Test();          t.Marks = 77;          Console.WriteLine("Marks : " + t.Marks);       }    } Which of the following lines can generate a compilation error?**

**1)**. Test t = new Test();

**2)**. **t.Marks = 77;**

**3)**. Console.WriteLine("Marks : " + t.Marks);

**4)**. **return mark;**

**Solution** :  
option [2] is correct

**Attempted** :  
option [4] is attempted

##### Q.**Which of the following is the correct way of collection initialization?**

**1)**. **List<int> digits = new List<int>{2, 4, 6, 8, 10};**

**2)**. List<int> digits = {2, 4, 6, 8, 10};

**3)**. List<int> digits {2, 4, 6, 8, 10};

**4)**. List<int> digits = {'a', 'b', 'c', 'd', 'e'};

**Solution** :  
option [1] is correct

**Attempted** :  
option [1] is attempted

##### Q.**Which of the following namespace allows to inspect type information and invoke methods in that type at runtime?**

**1)**. System.Attributes

**2)**. **System.Reflection**

**3)**. **System.Runtime.Serialization**

**4)**. System.Collections

**Solution** :  
option [2] is correct

**Attempted** :  
option [3] is attempted

##### Q.**The try block is followed by a finally block in which the \_\_\_\_\_\_\_\_\_ Finalize is explicitly called.**

**1)**. **Concrete Class**

**2)**. **Base Class**

**3)**. Derived Class

**4)**. Static Class

**Solution** :  
option [2] is correct

**Attempted** :  
option [1] is attempted

##### Q.**\_\_\_\_\_\_\_\_\_ block will be executed whether the exception occurs or not.**

**1)**. try

**2)**. catch

**3)**. throw

**4)**. **finally**

**Solution** :  
option [4] is correct

**Attempted** :  
option [4] is attempted

##### Q.**Which of the following statements are true? Statement 1: When user invoke the delegate, the instance method is called on the contained reference. Statement 2: User can add only one method reference to the invocation list of a delegate.**

**1)**. **Statement 1 is true but Statement 2 is not true**

**2)**. **Statement 2 is true but Statement 1 is not true**

**3)**. Both the statements are true

**4)**. Both the statements are false

**Solution** :  
option [1] is correct

**Attempted** :  
option [2] is attempted

##### Q.**Rashmi created an interface named IPrintable in her application with access specifier Protected Internal.  Where this interface will be available?**

**1)**. It will be available within current assembly only

**2)**. **It will be available withing current assembly or types derived from the containing class**

**3)**. It will be available within the containing type

**4)**. It will be available withing containing class or types derived from the containing class

**Solution** :  
option [2] is correct

**Attempted** :  
option [2] is attempted

##### Q.**Which of the following interface must be implemented when a class wants to sort its instances?**

**1)**. **IComparable**

**2)**. ICompare

**3)**. IList

**4)**. ISort

**Solution** :  
option [1] is correct

**Attempted** :  
option [1] is attempted

##### Q.**How many elements will be there in the following array? int [ , , ] arr = new int [3, 2, 3];**

**1)**. 8

**2)**. **12**

**3)**. **18**

**4)**. 29

**Solution** :  
option [3] is correct

**Attempted** :  
option [2] is attempted

##### Q.**Which of the following statements are NOT correct about C# classes?**

**1)**. Class is reference type

**2)**. **Object is a set of plans that specify how to build class**

**3)**. **Physical representation of class exist in memory when you just declare the object**

**4)**. **C# uses a class specification to construct objects**

**Solution** :  
option [2,3] are correct

**Attempted** :  
option [2,3,4] are attempted

##### Q.**Visual studio provides an object browser to enable developers to obtain info about the members of any class. This is an implementation of --------**

**1)**. strong names.

**2)**. Meta data.

**3)**. **Reflection.**

**4)**. **Extension methods.**

**Solution** :  
option [3] is correct

**Attempted** :  
option [4] is attempted

##### Q.**Which of the following is component used for exceptions that are usually thrown by the .NET runtime or  which are considered of a very generic nature and may be thrown by almost any application?**

**1)**. **SystemException**

**2)**. **DivideByZeroException**

**3)**. ApplicationException

**4)**. FormatException

**Solution** :  
option [1] is correct

**Attempted** :  
option [2] is attempted

##### Q.**Which of the following is not restriction on BinaryFormatter?**

**1)**. The class to be serialized must either be marked with the SerializableAttribute attribute, or must implement the ISerializable interface and control its own serialization and deserialization

**2)**. The binary format produced is specific to the .NET Framework and it cannot be easily used from other systems or platforms

**3)**. **The binary format is not human-readable, which makes it more difficult to work with if the original program that produced the data is not available**

**4)**. **It can serialize and restore only public members of an object**

**Solution** :  
option [4] is correct

**Attempted** :  
option [3] is attempted

##### Q.**Finalize is \_\_\_\_\_\_ method.**

**1)**. private

**2)**. **protected**

**3)**. **public**

**4)**. internal

**Solution** :  
option [2] is correct

**Attempted** :  
option [3] is attempted

##### Q.**\_\_\_\_\_\_\_\_\_ is used as listeners for events.**

**1)**. Event Handlers

**2)**. Iterators

**3)**. Functions

**4)**. **Delegates**

**Solution** :  
option [4] is correct

**Attempted** :  
option [4] is attempted

##### Q.**Match the following terms with its description:**

### **Column A**

1. Binary Formatter

2. Soap Formatter

3. Binary Serialization

4. Soap Serialization

### **Column B(Attempted)**

1. Can be used only for .NET Framework based application

2. Can be used by non .NET Framework based applications

3. class is generally not appropriate when data is meant to be passed through a firewall

4. class is primarily used for serialization through firewalls or among diverse systems

### **Solution**

1. class is generally not appropriate when data is meant to be passed through a firewall

2. class is primarily used for serialization through firewalls or among diverse systems

3. Can be used only for .NET Framework based application

4. Can be used by non .NET Framework based applications

##### Q.**What will be the output for the below given code? class Employee     {         public virtual void Show()         {             Console.WriteLine("Employee Class Show Method");         }     }     class Employee : Manager     {         public new void Manager()         {             Console.WriteLine("Manager Class Show Method");         }     }     class FooBarTest     {         static void Main(string[] args)         {             Employee obj = new Manager();             obj.Show();         }     }**

**1)**. **Employee Class Show Method**

**2)**. Manager Class Show Method

**3)**. **Compile time error occurs**

**4)**. Throws Exception at runtime

**Solution** :  
option [1] is correct

**Attempted** :  
option [3] is attempted

##### Q.**Which of the following is the correct syntax to create write only property?**

**1)**. class Maths  
    {  
        int num;  
  
        public writeonly int Num  
        {  
            get  
            {  
                return num;  
            }  
            set  
            {  
                num = value;  
            }  
        }  
    }

**2)**. **class Maths  
    {  
        int num;  
  
        public int Num  
        {  
            set  
            {  
                num = value;  
            }  
        }  
    }**

**3)**. **class Maths  
    {  
        public int Num  
        {  
            set  
            {  
                num = value;  
            }  
        }  
    }**

**4)**. class Maths  
    {  
        int num;  
  
        public Num  
        {  
            set  
            {  
                num = value;  
            }  
        }  
    }

**Solution** :  
option [2] is correct

**Attempted** :  
option [3] is attempted

##### Q.**Which of the following statement is correct about below code snippet? public class GenericClass<T> where T: class, IComparable { }**

**1)**. Class GenericClass requires that its type argument must implement IComparable interface

**2)**. **There are multiple constraints on type argument to GenericClass class**

**3)**. **Compile time error**

**4)**. Exception

**Solution** :  
option [2] is correct

**Attempted** :  
option [3] is attempted

##### Q.**Which of the following is not static class?**

**1)**. File

**2)**. Directory

**3)**. Path

**4)**. **DriveInfo**

**Solution** :  
option [4] is correct

**Attempted** :  
option [4] is attempted

##### Q.**Which of the following statements are true with respect to Garbage Collector? 1. Objects are destroyed more than once 2. Objects are destroyed while they are reachable 3. If the object overrides the Object.Finalize method then this method will be called by  the garbage collector.  4. You can not explicitly override Object.Finalize and even you can not explicitly call Finalize**

**1)**. 1 - True, 2 - True, 3 - True, 4 - True

**2)**. 1 - True, 2 - False, 3 - True, 4 - False

**3)**. **1 - False, 2 - False, 3 - True, 4 - False**

**4)**. **1 - False, 2 - False, 3 - True, 4 - True**

**Solution** :  
option [4] is correct

**Attempted** :  
option [3] is attempted

##### Q.**Which of the following statement/s is/are correct? Statement 1 : The Capacity property gives the total number of items stored in the ArrayList object Statement 2 : The Count property gets or sets the number of items that the ArrayList object can contain**

**1)**. Statement 1 is true but Statement 2 is false

**2)**. **Statement 2 is true but Statement 1 is false**

**3)**. Both statements are true

**4)**. **Both statements are false**

**Solution** :  
option [4] is correct

**Attempted** :  
option [2] is attempted

##### Q.**Which of the following symbol is used in the lambda expressions?**

**1)**. ->

**2)**. **=>**

**3)**. =

**4)**. -

**Solution** :  
option [2] is correct

**Attempted** :  
option [2] is attempted

##### Q.**Which of the following is the correct way to represent Lambda Expression?**

**1)**. (num1, num2) => int sum = num1 + num2; return sum;

**2)**. (num1, num2) => (int sum = num1 + num2; return sum;)

**3)**. **(num1, num2) => { int sum = num1 + num2; return sum; }**

**4)**. { (num1, num2) => int sum = num1 + num2; return sum; }

**Solution** :  
option [3] is correct

**Attempted** :  
option [3] is attempted

##### Q.**Identify the correct way to load the assembly and retrieve a list of various types in the assembly.**

**1)**. **Assembly maths = Assembly.LoadFrom("Maths.dll");  
Types[] mathType = maths.GetTypes();**

**2)**. **Assembly maths = Assembly.LoadFrom("Maths.dll");  
Types[] mathType = maths.GetType();**

**3)**. Assembly maths = Assembly.Load("Maths.dll");  
Types[] mathType = maths.GetTypes();

**4)**. Assembly maths = Assembly.Load("Maths.dll");  
Types[] mathType = maths.GetType();

**Solution** :  
option [1] is correct

**Attempted** :  
option [2] is attempted

##### Q.**What will be the output of the following code? 01 class Program 02   { 03       public int Print(var cities) 04       { 05          foreach (var i=0; i<cities.Length; i++)   06            { 07                Console.WriteLine(cities[i]); 08            } 09            return sum; 10        } 11        public static void Main() 12        { 13            string[] cities = { "Mumbai", "Pune", "Chennai" }; 14            new Program().Print(cities);  15        } 16    }**

**1)**. **Compilation error at line 3.**

**2)**. Compilation error at line 5.

**3)**. Execute successfully, but will not print anything

**4)**. Print output :   
Mumbai  
Pune  
Chennai

**Solution** :  
option [1] is correct

**Attempted** :  
option [1] is attempted

##### Q.**Which of the following statements are correct?**

**1)**. **Objects are dynamically allocated from a pool of free memory by using the new operator**

**2)**. **Garbage collector simplifies memory management**

**3)**. To create object, infinite memory is available

**4)**. **Garbage collector sets the managed heap pointer after the first object**

**Solution** :  
option [1,2] are correct

**Attempted** :  
option [1,2,4] are attempted

##### Q.**Check the following :  1. In the multicast delegate + operator is used to add methods to invocation list and to remove methods - operator is used. 2. In the multicast delegate methods will be invoked in the reverse manner, as they added. 3. Multicast delegates must contain only methods that return void.**

**1)**. **1 - True, 2 - False, 3 - True**

**2)**. **1 - True, 2 - False, 3 - False**

**3)**. 1 - False, 2 - False, 3 - False

**4)**. 1 - False, 2 - False, 3 - True

**Solution** :  
option [1] is correct

**Attempted** :  
option [2] is attempted

##### Q.**Async programming is used for \_\_\_\_\_\_**

**1)**. productivity

**2)**. **responsiveness**

**3)**. referencing

**4)**. dereferencing

**Solution** :  
option [2] is correct

**Attempted** :  
option [2] is attempted

##### Q.**Which of the following attribute is applied on the class to make an object available for serialization?**

**1)**. **[Serialization]**

**2)**. **[Serializable]**

**3)**. [Serialized]

**4)**. [Serialize]

**Solution** :  
option [2] is correct

**Attempted** :  
option [1] is attempted

##### Q.**Which of the following are the interfaces declared in System.Collections namespace?**

**1)**. **IComparer**

**2)**. **IEnumerable**

**3)**. **IDictionaryComparer**

**4)**. Icompare

**Solution** :  
option [1,2] are correct

**Attempted** :  
option [1,2,3] are attempted

##### Q.**Which of the following code will give the output "Number is Zero"?**

**1)**. delegate void NumericHandler(int num);  
  
    class NumericEvent  
    {  
        event NumericHandler MyNumericEvent;  
  
        public void InvokeNumberCheckHandler(int num)  
        {  
            if (MyNumericEvent != null)  
                MyNumericEvent(num);  
        }  
  
        public void NumberCheckHandler(int num)  
        {  
            if (num < 0)  
                Console.WriteLine("Number is Negative");  
            else if (num > 0)  
                Console.WriteLine("Number is Positive");  
            else  
                Console.WriteLine("Number is Zero");  
        }  
    }  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            NumericEvent evt = new NumericEvent();  
            evt.MyNumericEvent += evt.NumberCheckHandler;  
            evt.InvokeNumberCheckHandler(0);  
  
            Console.ReadKey();  
        }  
    }

**2)**. **delegate void NumericHandler(int num);  
  
    class NumericEvent  
    {  
        public event NumericHandler MyNumericEvent;  
  
        public void InvokeNumberCheckHandler(int num)  
        {  
            if (MyNumericEvent != null)  
                MyNumericEvent(num);  
        }  
  
        public void NumberCheckHandler(int num)  
        {  
            if (num < 0)  
                Console.WriteLine("Number is Negative");  
            else if (num > 0)  
                Console.WriteLine("Number is Positive");  
            else  
                Console.WriteLine("Number is Zero");  
        }  
    }  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            NumericEvent evt = new NumericEvent();  
            evt.MyNumericEvent += evt.NumberCheckHandler;  
            evt.InvokeNumberCheckHandler(0);  
  
            Console.ReadKey();  
        }  
    }**

**3)**. delegate void NumericHandler(int num);  
  
    class NumericEvent  
    {  
        public event NumericHandler MyNumericEvent;  
  
        public void InvokeNumberCheckHandler(int num)  
        {  
            if (MyNumericEvent != null)  
                MyNumericEvent(num);  
        }  
  
        public static void NumberCheckHandler(int num)  
        {  
            if (num < 0)  
                Console.WriteLine("Number is Negative");  
            else if (num > 0)  
                Console.WriteLine("Number is Positive");  
            else  
                Console.WriteLine("Number is Zero");  
        }  
    }  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            NumericEvent evt = new NumericEvent();  
            evt.MyNumericEvent += evt.NumberCheckHandler;  
            evt.InvokeNumberCheckHandler(0);  
  
            Console.ReadKey();  
        }  
    }

**4)**. delegate void NumericHandler(int num);  
  
    class NumericEvent  
    {  
        public event NumericHandler MyNumericEvent;  
  
        public void InvokeNumberCheckHandler(int num)  
        {  
            if (MyNumericEvent != null)  
                MyNumericEvent(num);  
        }  
  
        public void NumberCheckHandler(int num)  
        {  
            if (num < 0)  
                Console.WriteLine("Number is Negative");  
            else if (num > 0)  
                Console.WriteLine("Number is Positive");  
            else  
                Console.WriteLine("Number is Zero");  
        }  
    }  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            NumericEvent evt = new NumericEvent();  
                        evt.InvokeNumberCheckHandler(0);  
  
            Console.ReadKey();  
        }  
    }

**Solution** :  
option [2] is correct

**Attempted** :  
option [2] is attempted

##### Q.**Which of the following are correct ways to covert string to integer?**

**1)**. **int num;  
num = Convert.ToInt32(Console.ReadLine());**

**2)**. **int num;  
num = int.Parse(Console.ReadLine());**

**3)**. int num;  
num = Convert.float(Console.ReadLine());

**4)**. int num;  
num = int.Convert(Console.ReadLine());

**Solution** :  
option [1,2] are correct

**Attempted** :  
option [1,2] are attempted

##### Q.**Which of the following statement is not correct about constructors?**

**1)**. Constructor have the same name as there class name

**2)**. **Static constructor should have private access specifier**

**3)**. **Default constructor will get automatically created and invoked, if it is not specified in the class**

**4)**. Overloaded static constructors are not allowed in C#

**Solution** :  
option [2] is correct

**Attempted** :  
option [3] is attempted

##### Q.**What will be the output of below code snippet? class MyClassA {       public MyClassA()         {             Console.WriteLine("This is base class constructor");         } } class MyClassB : MyClassA {    } class Program {       static void Main(string[] args)         {             MyClassB obj = new MyClassB();         } }**

**1)**. **Prints : "This is base class constructor"**

**2)**. Code will execute successfully but no output

**3)**. Compile time error

**4)**. Exception

**Solution** :  
option [1] is correct

**Attempted** :  
option [1] is attempted

##### Q.**Radha is creating an application to accept Ticket details like name, age and quota.  The quota can take only the following values :General, Lady, Senior Citizen, Physically Challenged.  Which of the following data type is most suitable for quota?**

**1)**. class

**2)**. **enum**

**3)**. struct

**4)**. array

**Solution** :  
option [2] is correct

**Attempted** :  
option [2] is attempted

##### Q.**Which of the following classes can be used for reflection?**

**1)**. **Type.**

**2)**. **Reflection.**

**3)**. **Assembly.**

**4)**. **AssemblyName.**

**Solution** :  
option [1,3] are correct

**Attempted** :  
option [1,2,4] are attempted

##### Q.**What will be the output of below code snippet? abstract class A      {   public int i ;          public int j ;          public abstract void display();      }          class B: A          {   public int j = 20;          public override void display()           {   this.j = 15;              Console.WriteLine(i + "  " + j);          }      }          class Program      {   static void Main(string[] args)          {   B obj = new B();              obj.i = 10;              obj.display();               }      }**

**1)**. **10 15**

**2)**. **10 20**

**3)**. 0 15

**4)**. 0 20

**Solution** :  
option [1] is correct

**Attempted** :  
option [2] is attempted

##### Q.**Which of the following is the correct way to declare annonymous type?**

**1)**. **var person = new { Name = "Robert", Age = 35 };**

**2)**. Person person = new { Name = "Robert", Age = 35 };

**3)**. object person = new { Name = "Robert", Age = 35 };

**4)**. **Person person = new Person() { Name = "Robert", Age = 35 };**

**Solution** :  
option [1] is correct

**Attempted** :  
option [4] is attempted

Q.**Refer the below code snippet and complete the code                                                                 namespace Log4NetApp.Demo01  
{  
    public class Sample  
    {  
        private static readonly ILog log = LogManager.GetLogger("ConsoleLogger");  
  
        static void Main(string[] args)  
        {  
          \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
            log.Info("Entering application.");  
  
            for (int counter = 1; counter <= 10; counter++)  
            {  
                log.DebugFormat("Inside of the loop (Counter = {0})", counter);  
            }  
  
            try  
            {  
                throw new NotImplementedException("Testing log4net");  
  
            }  
            catch (NotImplementedException ex)  
            {  
                log.Fatal(ex.Message);  
            }  
  
            log.Info("Exiting application.");  
        }  
   }**

**1)**. **XmlConfigurator.Configure();**

**2)**. **XmlConfiguration.Configure();**

**3)**.     XmlConfigurator.Settings();

**4)**.     XmlConfiguration.Settings();

**Solution** :  
option [1] is correct

**Attempted** :  
option [2] is attempted

##### Q.**Which is the best place to close streams, data readers, connection objects used by your program?**

**1)**. Catch Block

**2)**. **Finally Block**

**3)**. Destructor

**4)**. Closing streams, data reader or connection object is not the responsibility of the developer.   
The framework takes care of this task.

**Solution** :  
option [2] is correct

**Attempted** :  
option [2] is attempted

##### Q.**State true or false: 1. Garbage collector releases the memory for objects that are being used 2. Garbage collector freeing up blocks of address space allocated to unreachable objects 3. Garbase collector sets the managed heap pointer after first object**

**1)**. **1 - False  
2 - True  
3 - False**

**2)**. 1 - True  
2 - True  
3 - True

**3)**. **1 - True  
2 - True  
3 - False**

**4)**. 1 - False  
2 - True  
3 - True

**Solution** :  
option [1] is correct

**Attempted** :  
option [3] is attempted

##### Q.**Sonali needs to handle error in unit testing which attribute you would recommend her?**

**1)**. TextFixture

**2)**. **ExpectedException**

**3)**. Setup

**4)**. TearDown

**Solution** :  
option [2] is correct

**Attempted** :  
option [2] is attempted

Q.**What will be the output of the following code?  
  
class Shape  
    {  
        public void Draw()  
        {  
            Console.WriteLine("Shape Draw Method");  
        }  
    }  
  
    class Square:Shape  
    {  
        public void Draw()  
        {  
            Console.WriteLine("Square Draw Method");  
        }  
    }  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            Shape sh = new Square();  
            sh.Draw();  
  
            Square sq = new Square();  
            sq.Draw();  
  
            Console.ReadKey();  
        }  
    }**

**1)**. Shape Draw Method  
Shape Draw Method

**2)**. **Shape Draw Method  
Square Draw Method**

**3)**. Square Draw Method  
Square Draw Method

**4)**. Shape Draw Method  
Square Draw Method  
Square Draw Method

**Solution** :  
option [2] is correct

**Attempted** :  
option [2] is attempted

##### Q.**public interface IPrintable {     void Print();     void PageSetup();  }  public interface IShow:IPrintable  {      void Show();      void Display();  } After interface inheritance, which methods will be member of IShow?**

**1)**. Show() and Display()

**2)**. **Print() and PageSetup()**

**3)**. **Print(), PageSetup(), Show() and Display()**

**4)**. Print(), PageSetup(), Show()

**Solution** :  
option [3] is correct

**Attempted** :  
option [2] is attempted

##### Q.**Which of the following is not property of Type?**

**1)**. **IsPrivate**

**2)**. **Namespace**

**3)**. FullName

**4)**. IsAbstract

**Solution** :  
option [1] is correct

**Attempted** :  
option [2] is attempted

##### Q.**Which of the following is not correct about Shared assemblies?**

**1)**. These are intended to be common libraries that any application can use

**2)**. **Shared assemblies are stored in Global Assembly Cache**

**3)**. **These assemblies do not require to maintain version information**

**4)**. A shared assembly must have strong name

**Solution** :  
option [3] is correct

**Attempted** :  
option [2] is attempted

##### Q.**Sunil has written the following code to define a Collection Initializer.  The code is using a Category class having ID and Name properties. public class CategoryList : IEnumerable     {         Category[] catlist = new Category[4];         public CategoryList()         {             catlist[0] = new Category(1001, "Accessories");             catlist[1] = new Category(1002, "Bikes");             catlist[2] = new Category(1003, "Cars");         }    } He would like to populate this data to List.  Which of the following is the correct way for the same?**

**1)**. **CategoryList clist = new CategoryList();  
List<Category> catList = new List<Category>();  
foreach (Category c in clist)  
catList.Add(c);**

**2)**. CategoryList clist = new CategoryList();  
List<Category> catList = new List<Category>();  
foreach (Category c in clist)  
catList.Add(c.Name);

**3)**. CategoryList clist = new Category();  
List<Category> catList = new List<Category>();  
foreach (Category c in clist)  
catList.Add(c);

**4)**. CategoryList clist = new Category();  
List<Category> catList = new List<Category>();  
foreach (Category c in clist)  
catList.Add(c.Name);

**Solution** :  
option [1] is correct

**Attempted** :  
option [1] is attempted

#### Section 5: Dot Net Framework

##### Q.**Check the following :  1. Type safety of the code is assured by CLR 2. CLR is a compiler for .NET Framework 3. CLR provides services to run unmanaged applications 4. CLR provides a code-execution environment that manages code targeting the .NET Framework**

**1)**. **1 - True, 2- False, 3 - False, 4 - True**

**2)**. 1 - False, 2- False, 3 - False, 4 - True

**3)**. 1 - True, 2- False, 3 - True, 4 - True

**4)**. 1 - True, 2- True, 3 - False, 4 - True

**Solution** :  
option [1] is correct

**Attempted** :  
option [1] is attempted

##### Q.**Check the following :  1. .NET class library is a collection of reusable classes, or types, that tightly integrate with the common language runtime. 2. The class library builds on the object-oriented nature of the runtime. 3. .NET class library reduces the learning curve associated with using a new piece of code. 4. Third-party components can not integrate with the classes in the .NET Framework.**

**1)**. 1 - True, 2 - True, 3- False, 4 - True

**2)**. **1 - True, 2 - True, 3- True, 4 - False**

**3)**. **1 - True, 2 - True, 3- False, 4 - False**

**4)**. 1 - True, 2 - True, 3- True, 4 - True

**Solution** :  
option [2] is correct

**Attempted** :  
option [3] is attempted

##### Q.**Which of the following is not correct about await?**

**1)**. The ‘await’ keyword is applied on the task in an asynchronous method and suspends the execution of the method, until the awaited task is not completed

**2)**. **await blocks the thread on which it is executing**

**3)**. **await signals to the compiler to continue with other async methods, till the task is in an await state**

**4)**. Once the task is completed, await resumes back where it left off from

**Solution** :  
option [2] is correct

**Attempted** :  
option [3] is attempted

##### Q.**What is the meaning of asterisk (\*) in AssemblyVersion (1.0.\*)?**

**1)**. **that the common language runtime will automatically provide default values for the last two numbers**

**2)**. there are NO values for the last two numbers

**3)**. that the common Type System will automatically provide default values for the last two numbers

**4)**. None of the above

**Solution** :  
option [1] is correct

**Attempted** :  
option [1] is attempted

##### Q.**MVC is introduced in which version of .NET Framework?**

**1)**. .NET Framework 3.0

**2)**. **.NET Framework 3.5**

**3)**. **.NET Framework 3.5 + SP1**

**4)**. .NET Framework 4.0

**Solution** :  
option [3] is correct

**Attempted** :  
option [2] is attempted

##### Q.**Which of the following library is introduced in .NET Framework 4.5?**

**1)**. **Task Based Async Model**

**2)**. **Task Parallel Library**

**3)**. LINQ

**4)**. DLR

**Solution** :  
option [1] is correct

**Attempted** :  
option [2] is attempted

##### Q.**What is the foundation of the .NET Framework?**

**1)**. BCL

**2)**. CLS

**3)**. CTS

**4)**. **CLR**

**Solution** :  
option [4] is correct

**Attempted** :  
option [4] is attempted

##### Q.**Which of the following component manages lifecycle of objects by reference counting and garbage collection?**

**1)**. **CLR**

**2)**. **CTS**

**3)**. CLS

**4)**. FCL

**Solution** :  
option [1] is correct

**Attempted** :  
option [2] is attempted

##### Q.**Which of the following statement is true? Statement 1: When user compiles CIL, native code is created, which is stored in an assembly. Statement 2: Assemblies include only executable application files.**

**1)**. Statement 1 is true but Statement 2 is not true

**2)**. **Statement 2 is true but Statement 1 is not true**

**3)**. Both the statements are true

**4)**. **Both the statements are false**

**Solution** :  
option [4] is correct

**Attempted** :  
option [2] is attempted

##### Q.**Single .NET DLL can contain \_\_\_\_\_\_ .NET laguages.**

**1)**. **One**

**2)**. Two

**3)**. Three

**4)**. Four

**Solution** :  
option [1] is correct

**Attempted** :  
option [1] is attempted

##### Q.**To answer drag and drop, correct term in answer area. 1. \_\_\_\_  provides a secure and versatile unit of processing that the CLR can use to provide isolation between applications. 2. If user developed an application which supports user interaction in English, French and Hindi,  then he should do language settings for this application in \_\_\_\_**

### **Attempted**

1. ApplicationDomain

2. Portable executables

### **Solution**

1. ApplicationDomain

2. Satellite Assemblies

3. Manifest

4. GlobalAssembly

5. Portable executables

##### Q.**Please provide the correct version number representation in .NET?**

**1)**. **<major version>.<minor version>.<build number>.<revision>**

**2)**. <major version>.< build number >.<revision>.< minor version >

**3)**. **<major version>.<minor version>.<revision>.< build number >**

**4)**. <major version>.<revision>.<build number>.< minor version >

**Solution** :  
option [1] is correct

**Attempted** :  
option [3] is attempted

##### Q.**\_\_\_\_\_\_\_\_\_\_\_ is a part of the execution process and involves metadata stored within the executable files.**

**1)**. **Class Loader**

**2)**. Class Manager

**3)**. **Code Loader**

**4)**. Code Manager

**Solution** :  
option [1] is correct

**Attempted** :  
option [3] is attempted

##### Q.**Drag and drop the correct answer from the given options: 1. The \_\_\_\_ manages code execution and used by the Class Loader to assign memory  for the objects and data. 2. \_\_\_\_ should be emitted by language compilers, to enable the runtime to provide  services to managed code.**

### **Attempted**

1. Code Manager

2. MSIL

### **Solution**

1. Code Manager

2. Metadata

3. COM Marshaler

4. MSIL

##### Q.**Which of the following statement/s is/are correct? Statement 1 : Intermediate Language is a CPU-independent set of instructions Statement 2 : Intermediate Language includes instructions for loading, storing, initializing, and calling methods on objects**

**1)**. Statement 1 is true but Statement 2 is false

**2)**. Statement 2 is true but Statement 1 is false

**3)**. **Both statements are true**

**4)**. Both statements are false

**Solution** :  
option [3] is correct

**Attempted** :  
option [3] is attempted

##### Q.**If user would like to look metadata and Common intermediate language (CIL)  code contained in a .NET Portable Executable (PE) file.  Which of the following tool from the .NET SDK should user use?**

**1)**. gacutil.exe

**2)**. **ildasm.exe**

**3)**. al.exe

**4)**. sn.exe

**Solution** :  
option [2] is correct

**Attempted** :  
option [2] is attempted

##### Q.**LINQ stands for \_\_\_\_\_\_**

**1)**. Linear Query

**2)**. **Language Integrated Query**

**3)**. Linear Integrated Query

**4)**. Language in Query

**Solution** :  
option [2] is correct

**Attempted** :  
option [2] is attempted

##### Q.**In command line compilation, following abbreviations are used for which compiler option?  To answer drag and drop correct keywords in answer area. /t : \_\_\_\_ /o : \_\_\_\_**

### **Attempted**

1. /temp

2. /out

### **Solution**

1. /target

2. /out

3. /temp

4. /objective

##### Q.**Which of the following contains all the metadata with assembly version, assembly scope,  reference to resources etc?**

**1)**. **Manifest**

**2)**. Metadata

**3)**. Resources

**4)**. **MSIL**

**Solution** :  
option [1] is correct

**Attempted** :  
option [4] is attempted

##### Q.**Which of the following is the preferred distribution and update mechanism in .NET Framework 4.5?**

**1)**. GAC

**2)**. **NuGet**

**3)**. JIT

**4)**. NGEN

**Solution** :  
option [2] is correct

**Attempted** :  
option [2] is attempted

##### Q.**Ashmita has created a class library named  ClientLogic   to which she wants to add  a strong name key and get the same registered in the GAC. What are the command that she needs to use on the Visual Studio command Prompt  to get this done?**

**1)**. **sn –k  ClientLogic.snk**

**2)**. **gacutil –i  ClientLogic.dll**

**3)**. gacutil –u ClientLogic

**4)**. sn –k  ClientLogic

**Solution** :  
option [1,2] are correct

**Attempted** :  
option [2] is attempted

##### Q.**Cross language interpretation provided by \_\_\_\_\_\_\_\_\_\_.**

**1)**. **Common Type System**

**2)**. **Common Language Specification**

**3)**. Reflection

**4)**. Attributes

**Solution** :  
option [1] is correct

**Attempted** :  
option [2] is attempted