C# 3.5

TIPS:

1.No ‘n+1’ and ‘n-1’ concept will work out. So please check the exact number of options they ask to choose.

2.Read Questions properly.

1. Which statements are true about yield statements? (select 4)
   a. A yield statement can also appear in an anonymous method.
   b. A yield statement cannot appear in an anonymous method.
   c. When used with expression, a yield return statement cannot appear in a catch block or in a try block that has one or more catch clauses.
   d. A yield statement can only appear inside an iterator block.
   e. The yield keyword signals to the compiler that the method in which it appears is an iterator block.

Answer: - b, c, d, e

2. Extension methods gives you the capability of adding methods of existing types of our own types, without creating the new derived class.

   Answer: - True (These are the special type of static methods)

3. Query expressions can be compiled to expression trees or to delegates.

   Answers: - True

4. A generic type parameter that is not marked covariant or contra variant is reference to as_____.
   a. Nonvariant
   b. Variant
   c. Neutral
   d. Invariant

   Answer: - d (http://msdn.microsoft.com/en-us/library/dd799517.aspx)

5. Generic keyword is used for covariance and contra variance.

   Answer: - True

6. How to get the type of assembly?

Answer: - GetType()

7. How to define dynamic module of an assembly?

   Answer: - DefineDynamicModule(string)

8. var keyword is use to initialize –

   Answer: - Object

9. Which is the auto-increment able (http://msdn.microsoft.com/en-us/library/system.collections(v=vs.71).aspx)

   Answer: - ArrayList()

10. You can use the _____ delegate to represent a method can be passed as a parameter, without explicitly declaring a custom delegate.

    Answer: - Func(Of TResult) delegate

11. List<int> digits= new List<int>{0,1,2,3,4,5,6,7,,8,9};

    Answer:- Collection Intializer

12. Specify condition when partial classes become necessary

    Answer: -
    a. When working on large projects, spreading a class over separate files allows multiple programmers to work on it simultaneously.
    b. When working with automatically generated source, code can be added to the class without having to recreate the source file.
    c. To split a class definition, use the partial keyword modifier.

13. void Swap<T>(List<T> list1, List<T> list2)
    {
        //code to swap items

```
}
void Swap(List<int> list1, List<int> list2)
{
        //code to swap items
}
```

Answer: - Open constructed and closed constructed types can be used as method parameters

14.Difference b/w action<T> and func<TResult>

Answer: - action<t> Encapsulates a method that has a single parameter and does not return a value.

Func(TResult:)Encapsulates a method that has no parameters and returns a value of

the type specified by the *TResult* parameter.

NOTE:

--->Main point to remember is that: Action does not return value and Func returns value.

->it s Func(TResult) and not Function(TResult)

--->Example: IF Func(T,TResult):Has a single parameter(the parameter count is based on no of 'T' specified) and returns a value in TResult.

15.A flexible and secure method of isolating running applications.

Answer:- appdomain()

16.Generic and Non-Generic inherited from base class

Answer: - True

17.How non generic, generic is inherited from base class.

Answer: -

1. Generic classes can inherit from concrete, closed constructed, or open constructed base classes

2. Non-generic —can inherit from concrete, closed constructed base classes, but not from open constructed classes

18.What are true about query expression?

Answer:-

a. Query expressions are often more readable than equivalent expressions written in
b. Some query operations, such as Count or Max, have no equivalent query expression clause and must therefore be expressed as a method call
c. Query expressions can be compiled to expression trees or to delegates

19.True about anonymous method.

Answer:-

Some option with: -- Use jump,goto ,break it is compiler error or not.

30. How to Creates an instance of the specified type 'whose name is specified' using the constructor

Answer:- CreateInstance()

31.var V item =new (Rate =90, msg="hello");

Answer: -

1.Example of anonymous type

2. anonymous type that is initialized with two properties named Rate and msg.

32. _____ Method locates the specified type from this assembly & creates instance using system.

Answer: - Create Instance()

34.Question on anonymous type:

Answer:-

a. If two or more anonymous types in the same assembly have the same number and type of properties, in the same order, the compiler treats them as the same type
b. They share the same compiler-generated type information.

36. On nested types

```
public class Container
{
   public class Nested
   {
      private Container parent;

      public Nested()
      {
      }
      public Nested(Container parent)
      {
         this.parent = parent;
      }
   }
}
```

Answer:- Ans is within this (read it from msdn type the que):

A nested type has access to all of the members that are accessible to its containing type. It can access private and protected members of the containing type, including any inherited protected members.
the full name of class Nested is Container.Nested. This is the name used to create a new instance of the nested class, as follows
The nested, or inner type can access the containing, or outer type. To access the containing type, pass it as a constructor to the nested type
A type defined within a class or struct is called a nested type

Regardless of whether the outer type is a class or a struct, nested types default to private, but can be made public, protected internal, protected, internal, or private. In the previous example, Nested is inaccessible to external types, but can be made public like this:
The nested, or inner type can access the containing, or outer type. To access the containing type, pass it as a constructor to the nested type

Nested types can access private and protected members of the containing type, including any inherited private or protected members.

37.Var keyword is used for

Answer:-

- var can only be used when a local variable is declared and initialized in the same statement; the variable cannot be initialized to null, or to a method group or an anonymous function.
- var cannot be used on fields at class scope.
- Variables declared by using var cannot be used in the initialization expression. In other words, this expression is legal: int i = (i = 20); but this expression produces a compile-time error: var i = (i = 20);
- Multiple implicitly-typed variables cannot be initialized in the same statement.
- If a type named var is in scope, then the var keyword will resolve to that type name and will not be treated as part of an implicitly typed local variable declaration.

You may find that var can also be useful with query expressions in which the exact constructed type of the query variable is difficult to determine. This can occur with grouping and ordering operations.
The var keyword can also be useful when the specific type of the variable is tedious to type on the keyboard, or is obvious, or does not add to the readability of the code. One example where var is helpful in this manner is with nested generic types such as those used with group operations. In the following query, the type of the query variable is IEnumerable<IGrouping<string, Student>>. As long as you and others who must maintain your code understand this, there is no problem with using implicit typing for convenience and brevity

38. LINQ to DataSet
Given some query.So choose two options involving the keyword dataset

39..LINQ to XML
Given some query with some 'load' method.Choose option involving key words,
1.linq to xml and
2.element containing attribute value.

40. The preference for generic classes is to use generic interfaces ,such asTComparable<T>rather than IComparator,why??
To avoid boxing and unboxing operations on value type.

41. Unsafe code can be accessed within the anonymous-method-block. State true or false.
Ans: False

42. Which method gets serialization information with all of the data needed to reinstantiate this assembly?
a. GetObjectData
b. GetFile
c. GetModule
d. GetReferencedAssemblies
Ans: a

43. What are iterators? Select three correct answer options.
a. An iterator can be used as the body of a methods, an operator, or a get accessor.
b. An iterator can be used as the body of a structure, a class, or a set accessor.
c. It is a section of code that returns an ordered sequence of values of the same type.
d. The iterator code uses the yield return statement to return each element in turn, yield break ends the iteration.
e. It is a section of code that returns an ordered sequence of values of the different types.
Ans: a,c,d

44. _____ class from the System.Collections namespace Implements the IList interface using an array whose size is dynamically increased as required
a. HashTable
b. ArrayList
c. Queue
d. Array
Ans: b

45. You can use the _____ delegate to represent a method that can be passed as a parameter, without explicitly declaring a custom delegate.
a. Func<TResult>
b. Action
Ans: a

46. What is the significance of DynamicMethod class? Select three.
a. It defines and represents a dynamic method that can be compiled, executed and stored.
b. It defines and represents a dynamic method that can be compiled, executed and discarded.
c. Discarded methods are available for garbage collection.
d. It can be used to generate and execute a method at run time, with the dynamic assembly getting generated automatically.
e. It can be used to generate and execute a method at run time, without having to generate a dynamic assembly.
Ans: b,c,e

47. A generic interface or generic delegate type can have both covariant and contravariant type parameters.
Ans: True

48. What does the following code denote?
Public static iEnumerable<TResult> join<TOuter,TInner,TKey,TResult>(
This IEnumerable<TOuter> outer,
IEnumerable<TInner> inner,
Func<TOuter,Tkey> outerKeySelector,
Func<TInner,Tkey> innerKeySelector,
Func<TOuter,TInner,Tresult> resultSelector);
)
a. Usage of Projection Operator
b. Usage of Concatenation Operator
c. Usage of Join Operator
d. Usage of Grouping Operator
Ans: c

49. When an anonymous type is assigned to a variable, that variable must be initialized with the _____ construct.
a. object
b. init
c. value
d. var
Ans: d

50. What are the statements true about XML Serialization?
a. XML Serialization does not include methods, indexers, private fields, or read-only properties (except read-only collections)
b. XML Serialization serializes the public fields and properties of a class, or the parameters and return values of methods into an XML stream.
c. XML serialization is used to serialize all an object's fields and properties, both public and private.
d. System.Serialization is used to perform XML serialization and deserialization.
e. XML is not an open standard, the resulting XML stream can not be processed by any application or any platform.
Ans: a,b

51. All the type parameters of which interfaces are covariant?
a. IComparable<T>
b. IEnumerator<T>
c. IQueryable<T>

d. IComparer<T>
e. IEnumerable<T>
Ans: b,c,e

52. What are your observation on the following code? Select three.

```
public class Container
{
public class NClass
{
private Container parent;
public NClass()
{
}
public NClass(Container parent)
{
this.parent=parent;
}
}
}
```

a. Following code is an example of a Nested class.
b. In the code the inner type can access the containing type by passing it as a constructor to the inner type.
c. The Nested type can access the private and protected members of the containing type.
d. In the code the containing type can access the inner type by passing it as a constructor to the containing type.
e. Following code is an example of a Derived class.
Ans: a,b,c

53. Generic types can use multiple type parameters and constraints. State true and false.
Ans: true

54. Extension methods give you the capability of adding methods of existing types to our own types, without creating a new derived class. State true and false.
Ans: True

55. What does the following code denote? Select three.

```
List<int> numbers=new List<int>() {5,4,1,3,9,8,6,7,2,0};
IEnumerable<int> filteringQuery=
From num in numbers
Where num<3||num>7
Select num;
```

a. This shows how to query the Database.

b. This shows how to filter or restrict results by applying conditions with a where clause.
c. It returns all elements in the source sequence whose values are greater than 7 or less than 3.
d. This code is an example of LINQ query.
e. This code shows how to Order the data by grouping the numbers in the list.
Ans: b,c,d

56. _____ method locates the specified type from this assembly and creates an instance of it using the system activator.
a. GetTypes
b. GetName
c. CreateInstance
d. DefineResource
Ans: c

57. What does the following code snippet denote? Select three.
Public string Name{ get; set; }
a. No field declaration, no code to get and set the value of the field is required.
b. It denotes implementation of Anonymous Properties.
c. Field declaration, get and set values are added to the property when the application is loaded.
d. It denotes implementation of Automatic Properties.
e. Field declaration, get and set values can be added to the property in the future.
Ans: a,b,c

58. Which statements are true about Lambda Expression? Select Three.
a. The left side of the lambda operator specifies the input parameters and the right side holds the expression or statement block.
b. All lambda expressions use the lambda operator !>
c. A lambda expression is an anonymous function that can contain expressions and statements, and can be used to create delegates or expression tree types.
d. The left side of the lambda operator holds the expression or statement block and the right side specifies the input parameters.
e. All lambda expressions use the lambda operator =>
Ans: a,c,e

59. What are your observations on the following code?
```
Using System.Linq.Expressions;
Namespace ConsoleApplication1
{
Class Program
{
```

```
Static void Main(string[] args)
{
Expression<del> myET = x => x*x;
}
}
}
```

a. In the example lambda expression creates an anonymous method.
b. In the example lambda expression creates an expression tree type.
c. In the example lambda expression is assigned to a delegate type.
d. In the example lambda expression creates an anonymous type.
Ans: b

60. Which of the following statements are true abount the new() constraint in generics? Select three.
a. The new constraint specifies that any type argument in a generic class declaration must have a public parameterless constructor.
b. Apply the new constraint to a type parameter, when your generic class creates new instances of the type.
c. When you use the new() constraint with other constraint, it must be specific first.
d. When you use the new() constraint with other constraints, it must be specified last.
e. The new constraint specifies that any type argument in a generic class declaration must have a private parameterless constructor.
Ans: a,b,d

61. In a generic type definition, the _____ clause is used to specify constraints on the types that can be used as arguments for a type parameter, defined in a generic declaration.
a. for
b. where
c. group
d. if
Ans: b

62. Which statement are true about AssemblyBuilder Class? Select three.
a. Extends an existing assembly.
b. Defines and represents a dynamic assembly.
c. To generate an executable the SetEntryPoint method must be called to identify the method that is the entry point to the assembly.
d. The extended modules in the existing assembly are saved using the Save method.
e. the dynamic modules in the assembly are saved when the dynamic assembly is saved using the Save method.
Ans: b,c,e

63. What are the statements are true about Extension Methods? Select two.
a. Extension method are only in scope when you explicitly import the namespace into your source code with “using” directive.
b. You can extend standard interfaces with additional methods by physically altering the existing class libraries.
c. You can extend .NET types and older COM/ActiveX control types & use them in old applications through Extension methods.
d. Extension method does not allow you to add new method to the existing class.
e. Extension method are special kind of static method but they are called using instance method syntax.
Ans: a,e

64. What does the var keyword do?
a. The var keyword instructs the complier to infer the type of the variable from the expression on the right side of the initialization statement.
b. The var keyword instructs the compiler to infer the type of the variable as integer.
c. The var keyword instructs the compiler to infer the type of the variable based on the return value of the method in which the variable resides.
d. The var keyword instructs the compiler to infer the type of the variable as object.
Ans: d

65. What does the following code snippet denote? Select two.

```
Public partial class Manufacturer
{
Public void ManufactureItem()
{
}
}
Public partial class Manufacturer
{
Public void sellItems()
{
}
}
```

a. Each partial class holds a different method.
b. The second partial class is contained in the first class.
c. Following code denotes usage of a partial class.
d. Following code denotes usage of an abstract class.
Ans: a,c

66. What does the code denote? Select two.

```
Var vTemp = new { Rate = 97, Message = “New Rate” };
```

a. The following example shows the anonymous type being initialized with two properties – Rate & Message.
b. Following code is an example of anonymous type.
c. Following code is an example of a structure of type var.
d. The following example shows the structure being initialized with two properties – Rate & Message.
Ans: a,b

67. How does the compiler react, if two or more anonymous types have the same number and type of properties in the same order?
a. The compiler treats them as the same type and they share the same compiler-generated type information.
b. The compiler overrides each anonymous type.
c. The compiler overloads each type with a dummy parameter and they share different compiler-generated type information.
d. The compiler adds an incremental numeric suffix to each anonumous type & they share different compiler-generated type information.
Ans: a

68. What does the following code denote?
String[] groupingQuery = {“carrots”, “cabbage”, “broccoli”, “beans”, “barley”};
IEnumerable<IGrouping<char, string>> queryFoodGroups =
From item in groupingQuery
Group item by item[0];
a. This query expression demonstrates how to filter or restrict results by applying conditions with a where clause.
b. This code is an example of LINQ query and Shows how to query the Database.
c. This Shows how to order the returned results.
d. This Shows how to group results according to a key and returns two groups based on the first letter of the word.
e. This Shows how to filter or restrict results by applying results by applying conditions with a where clause.
Ans: d

69. What are the statements that are not true about Anonymous types? Select two.
a. Anonymous types are Value types and can not contain read-only properties.
b. Anonymous types typically are used in the select clause of a query expression to return a subset of the properties from each object in the source sequence.
c. Anonymous types are class types that can be cast to any easily.
d. Anonymous types contain one or more public read-only properties.
Ans: a,c